GALACTIC: Global and Local Agnostic Counterfactuals for Time-series Clustering

Christos Fragkathoulas
ch.fragkathoulas@athenarc.gr
University of Ioannina
Ioannina, Greece
Archimedes, Athena Research Center
Athens, Greece

Eleni Psaroudaki
h.psaroudaki@athenarc.gr
Archimedes, Athena Research Center
Athens, Greece

Themis Palpanas
themis@mi.parisdescartes.fr
Université Paris Cité, LIPADE, F-75006
Paris, France
Archimedes, Athena Research Center
Athens, Greece

Evaggelia Pitoura
pitoura@uoi.gr
University of Ioannina
Ioannina, Greece
Archimedes, Athena Research Center
Athens, Greece

arXiv:2603.05318v2 [cs.LG] 9 Mar 2026

Abstract

Time-series clustering is a fundamental tool for pattern discovery, yet existing explainability methods, primarily based on feature attribution or metadata, fail to identify the transitions that move an instance across cluster boundaries. While Counterfactual Explanations (CEs) identify the minimal temporal perturbations required to alter the prediction of a model, they have been mostly confined to supervised settings. This paper introduces GALACTIC, the first unified framework to bridge local and global counterfactual explainability for unsupervised time-series clustering. At instance level (local), GALACTIC generates perturbations via a cluster-aware optimization objective that respects the target and underlying cluster assignments. At cluster level (global), to mitigate cognitive load and enhance interpretability, we formulate a representative CE selection problem. We propose a Minimum Description Length (MDL) objective to extract a non-redundant summary of global explanations that characterize the transitions between clusters. We prove that our MDL objective is supermodular, which allows the corresponding MDL reduction to be framed as a monotone submodular set function. This enables an efficient greedy selection algorithm with provable $(1 - 1/e)$ approximation guarantees. Extensive experimental evaluation on the UCR Archive demonstrates that GALACTIC produces significantly sparser local CEs and more concise global summaries than state-of-the-art baselines adapted for our problem, offering the first unified approach for interpreting clustered time-series through counterfactuals.

Keywords

Counterfactual Explanations, Clustering, Explainability, Time-Series

1 Introduction

Clustering is a core tool for analyzing time-series data across domains where large collections of temporal signals must be grouped to reveal shared structure without supervision. Applications span from finance [15], neuroscience [16], robotics [27], biology [36], speech processing [25], multimedia [12], and critical infrastructure [4]. In such settings, clustering acts as a critical dimensionality reduction step, transforming massive volumes of temporal signals into a manageable set of behavioral regimes. However, the utility of these clusters is often bottlenecked by their opacity. Practitioners rarely need just a label; they require an understanding of *why* a series belongs to a specific cluster and *how* its behavior must evolve to transition to one. Without such actionable insights, clusters remain descriptive artifacts rather than decision-supporting tools.

Unfortunately, time-series clustering pipelines are frequently opaque in practice. Whether due to proprietary logic, complex non-linear distance metrics (e.g., DTW), or multi-stage feature transformations, the effective assignment logic is often inaccessible [8, 33]. Current explainability research broadly follows two paths: *inherently interpretable models* and *post-hoc surrogates*. The former is often architecture-specific, embedding explanations into the clustering model itself [e.g., 7, 9]. The latter typically provides only static summaries or surrogate models [e.g., 6, 28] that fail to offer a unified path from local instance-level reasoning to compact, global cluster-level insights.

Counterfactual explanations (CEs) offer a compelling solution by identifying minimal, actionable changes to an input that induce a label flip [39]. While in time-series classification, several CE frameworks have been proposed [e.g., 2, 3, 19, 22], existing methods often struggle with temporal coherence, frequently overwrite characteristic structure, and fail to localize changes to truly discriminative regions [3, 19, 22]. More importantly, the existing methods that are designed for supervised labels do not address the unsupervised nature of discovering transitions between latent time-series clusters or the combinatorial challenge of distilling thousands of local counterfactuals into a concise global summary.

We address these gaps with Galactic, *a unified framework for local and global counterfactual explainability in time-series clustering, agnostic to the underlying clustering procedure.* At the *local* level, Galactic employs an importance-guided gradient search that restricts edits to discriminative temporal regions identified via subgroup-level permutation analysis. This projection-based optimization ensures that local explanations are not only low-cost but also structurally preserved, respecting the characteristic shape of the underlying time-series.

At the *global* level, we move beyond isolated instances to identify *perturbations*: adjustment patterns that explain transitions for entire populations. While traditional global recourse methods (for supervised models on tabular data) yield incomparable Pareto fronts that require manual trade-off selection due to the multiobjective nature of the problem [14, 20, 23], Galactic reformulates global selection as an Information-Theoretic Model Selection problem. By employing the *Minimum Description Length (MDL)* principle [31], we transform the competing goals of coverage and complexity into a single, principled objective. MDL automatically identifies the "optimal" summary by selecting a non-redundant set of perturbations that maximizes explanatory power without arbitrary weight tuning.

Furthermore, we prove that our MDL objective behaves as a monotone submodular set function. This allows us to provide efficient greedy selection strategies with $(1 - 1/e)$ approximation guarantees, alongside a scalable hierarchical refinement mechanism that adapts to the internal density and complexity of cluster subgroups. To summarize, our *contributions* are:

(1) **Problem formulation.** We formalize counterfactual explainability for time-series clustering at both *instance* (local) and *cluster/subgroup* (global) level in a model-agnostic setting.

(2) **Local counterfactuals with structural constraints.** We introduce a constrained gradient optimization that utilizes subgroup and cluster specific temporal importance to produce sparse, shape-preserving counterfactuals.

(3) **Global counterfactual summaries via MDL.** We propose a parameter-free model selection objective based on the MDL principle for identifying the set of global perturbations, resolving the traditional multi-objective trade-off. We characterize the submodularity of the MDL objective, deriving a greedy selection strategy with guaranteed approximation bounds.

(4) **Adaptive Hierarchical Refinement.** We develop an incremental refinement mechanism that utilizes the underlying density of cluster subgroups to provide tunable global explanations. This allows for a variable number of counterfactuals that adapt to the internal structural complexity of each partition and also allows for high-dimensional scalability.

(5) **Thorough Experimental Evaluation.** We demonstrate that Galactic produces sparser local CFs and more concise, higher-coverage global summaries than adapted state-of-the-art baselines.

The paper structure is: Section 2 summarizes the related work; Section 3 introduces the local and global problem for counterfactual explainability in time-series clustering; Section 4 presents Galactic framework; Section 5 presents the experimental evaluation; and finally, Section 6 concludes with future work.

2 Related Work

The pursuit of explainability in time-series clustering has followed two distinct paths: intrinsic model design and post-hoc explanations. Intrinsic methods aim to create "self-explanatory" clusters by integrating interpretability directly into the architecture. For instance, Bertsimas et al. [7] utilize hierarchical clustering paired with decision trees to optimize variable importance, while T-DPSOM [24] leverages variational autoencoders and self-organizing maps to maintain a topology-preserving latent space. Other approaches, such as k-Graph [9] and Time2Feat [8], identify "graphoids" or a rich set of spectral and temporal features to characterize clusters.

Despite these advances, most applications rely on post-hoc explanation methods, which interpret black-box cluster assignments after they are formed. Early efforts in this space focused on metadata ranking [6, 32] or training surrogate classifiers to apply local interpretability tools like SHAP or Grad-CAM [28]. However, as noted by Ntekouli et al. [26], these methods often suffer from a "localization gap". While they can identify which variables are globally important, they frequently fail to provide evidence of specific temporal segments or timesteps that drive cluster membership, often collapsing into broader feature-level summaries rather than precise temporal insights.

Counterfactual Explanations (CEs) offer a more prescriptive alternative by identifying minimal changes needed to alter a prediction. While the foundational work by Wachter et al. [39] introduced gradient-based optimization for this purpose, in its time-series adaptation, it lacked temporal awareness, leading to "noisy" or non-contiguous edits. The field has since evolved toward localized and sparse edits. TSEvo [19] employs a multi-objective evolutionary algorithm to optimize for proximity, sparsity, and plausibility using segment-based crossovers. Glacier [40] introduces locally constrained counterfactuals where edits are guided by constraint vectors to favor specific intervals. Similarly, CEI [41] constrains edits to a fixed number of contiguous temporal intervals, using a combinatorial identity to enforce interval sparsity. While these methods, alongside others like Sub-SpaCE [30] and Native Guide [11], achieve high local fidelity in supervised classification, they are not designed for the unsupervised nature of clustering. Current counterfactual clustering explanations are often tied to specific algorithmic architectures [e.g., 37] and lack the specialized operators required for the temporal dependencies of time-series data [35].

Galactic bridges the gap between time-series counterfactual search and the unsupervised setting of clustering by operating on cluster assignments queried through a surrogate. Furthermore, while these existing frameworks are focused on the instance (local) level, Galactic also introduces a cluster (global) level approach. By framing the selection representative temporal "perturbations" as a Minimum Description Length (MDL) optimization problem, it provides a unified approach that reveals the temporal structure of clusters across multiple granularities (instance, subgroup, cluster). At the global level, we generate a set of local CFEs as candidate perturbations cluster/subgroup-level temporal adjustment patterns, and formulate their selection as a Minimum Description Length optimization problem. Overall, this yields concise explanations at multiple granularities (cluster, subgroup, and instance), revealing which temporal regions matter and what minimal adjustments shift meaningful portions of a cluster toward others.

3 Problem Definition

3.1 Preliminaries

Let $\mathcal{T} = \mathbb{R}^T$ denote the space of univariate time series of fixed length T. An instance $\boldsymbol{x} \in \mathcal{T}$ is represented as a sequence of temporal observations $\boldsymbol{x} = [x_1, \ldots, x_T]$. We model x as a concatenation of M contiguous, non-overlapping segments $\mathcal{S}(x) = \{I_1, \ldots, I_M\}$

defined by breakpoints $0 = \tau_0 < \tau_1 < \cdots < \tau_M = T$ such that $I_m = \{\tau_{m-1} + 1, \ldots, \tau_m\}$.

Given a dataset $X = \{\boldsymbol{x}^{(i)}\}_{i=1}^N \subset \mathcal{T}$, we consider a clustering algorithm $f : \mathcal{T} \to \{1, \ldots, K\}$ that maps each instance to one of K discrete clusters. The algorithm induces a partition $C = \{C_1, \ldots, C_K\}$ of the dataset X, where each cluster C_k is defined as the set of instances assigned to index k:

$$C_k = \{x^{(i)} \in X \mid f(x^{(i)}) = k\}$$

By definition, these clusters are disjoint ($C_i \cap C_j = \emptyset$ for $i \neq j$) and exhaustive ($\bigcup_{k=1}^K C_k = X$).

3.2 Probabilistic Surrogate of Cluster Assignments

While some clustering algorithms provide an explicit decision rule, many state-of-the-art methods, particularly spectral and graph-based approaches, are *transductive*, meaning they lack an out-of-sample mapping function f without requiring model retraining (or learning an explicit extension) [1, 5, 38]. Even for inductive methods, the underlying logic may be a "black-box" providing only a discrete partition C on X. To maintain a model-agnostic framework, we frame the explanation task as the problem of explaining the observed assignments by learning a probabilistic surrogate classifier $g : \mathcal{T} \to [0, 1]^K$. Here, $g_k(\boldsymbol{x})$ denotes the predicted probability that instance $\boldsymbol{x}$ belongs to cluster C_k. We define the predicted assignment as $\arg\max_{k \in [K]} g_k(\boldsymbol{x})$, which we denote simply as $\arg\max g(\boldsymbol{x})$.

We frame this as a supervised learning problem on the induced partition C. Specifically, we optimize the parameters of g to maximize the log-likelihood of the observed assignments $f(\boldsymbol{x}^{(i)})$ over X via a categorical cross-entropy objective:

$$\max_g \frac{1}{N} \sum_{i=1}^{N} \sum_{\mathbf{k}=1}^{\mathbf{K}} \mathbb{1}\{\mathbf{f}(\mathbf{x}^{(\mathbf{i})}) = \mathbf{k}\} \log(\mathbf{g}_{\mathbf{k}}(\mathbf{x}^{(\mathbf{i})})),$$

where $g(\boldsymbol{x}^{(i)}) \in \Delta^{K-1}$ is the predicted probability distribution over clusters, subject to the surrogate achieving high empirical accuracy:

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{\arg\max_{k \in [K]} g(\boldsymbol{x}^{(i)}) = f(\boldsymbol{x}^{(i)})\} \approx 1.$$

Crucially, because f is often defined only over the specific distribution of X, the surrogate g is required only to capture the local decision boundaries within the existing data manifold. We make no claim of generalization to out-of-distribution samples; instead, the surrogate serves as a differentiable or queryable proxy that enables the optimization of counterfactual explanations.

3.3 Local Counterfactual Explanations

For an instance $x \in C_k$, a **counterfactual explanation** $\boldsymbol{x}^{cf}$ is a perturbed instance $\boldsymbol{x}^{cf} = \boldsymbol{x} + \boldsymbol{\delta}$, where $\boldsymbol{\delta} \in \mathbb{R}^T$ represents a **minimal perturbation** (see e.g., Figure 1) such that the cluster assigment of $\boldsymbol{x}^{cf}$ differs from the original, i.e., $\arg\max g(\boldsymbol{x}^{cf}) = k'$ where $k' \neq k$. We refer to k' as the target cluster. In our framework, the target cluster k' is identified as the most probable alternative assignment according to the posterior distribution of the surrogate: $k' = \arg\max_{c \neq k} g(x)$. Formally, x^{cf} is the solution to an optimization problem that balances cluster validity and perturbation cost.

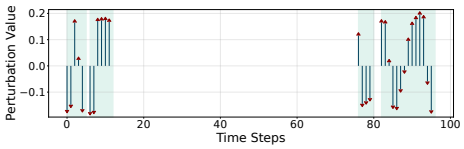


Figure 1: Example of a Perturbation $\boldsymbol{\delta}$

The **perturbation cost** $\text{cost}(\boldsymbol{x}, \boldsymbol{x}^{cf})$ can be defined as any distance metric, such as ℓ_p. By optimizing this objective, we identify the minimal temporal adjustment needed to cross the decision boundary of the surrogate model.

Problem 1 (Local Level Countrerfactuals). *Given an instance $\boldsymbol{x} \in X$ and its surrogate assigment distribution $g(\boldsymbol{x})$ with $\arg\max\ g(x) = k$, we seek the optimal counterfactual $\boldsymbol{x}^{cf}$ by minimizing the cost subject to a cluster-flip constraint:*

$$x^{cf} = \arg\min_{x'} cost(x, x') \quad s.t. \quad \arg\max_{c \in [K]} g(x') = k', \quad k' \neq k$$

Desired properties for time-series clustering counterfactual explanations. A high-quality local counterfactual for time series must satisfy *two fundamental criteria* to ensure both interpretability and utility. First, it must maintain ***proximity*** [39], remaining as close as possible to the original instance under a chosen distance function to ensure the perturbation is minimal. Second, it must exhibit ***sparsity*** [39] by altering only the few critical timesteps necessary to induce a change in the assignment of the surrogate.

Finally, in the *specific context of clustering*, the explanation must additionally demonstrate ***structural relevance*** [9]. Unlike supervised counterfactuals that merely seek any valid flip, clustering explanations must align with the underlying temporal patterns that define cluster membership. This requires focusing perturbations on the most discriminative segments of the time series. These high-importance regions are the characteristic temporal patterns that determine why an instance belongs to its cluster. By prioritizing changes in these segments, the counterfactual preserves the global temporal structure while highlighting the specific local features responsible for the cluster assignment.

3.4 Global Counterfactual Explanations

While local explanations explain why a specific time-series was assigned to a cluster, global counterfactual explanations provide a concise summary of how the instances of an entire cluster C_k can cross the cluster boundaries. The global explanation problem is often framed as a multi-objective optimization task [20] and can be defined as *the selection of a subset $\mathbb{S} \subseteq \mathbb{A}_k$ of perturbations that provide a valid recourse (cluster flip) for the largest possible number of time-series (effectiveness) while maintaining minimal cumulative cost (average flipping cost).* $\mathbb{A}_k$ represents the possibly infinite set of all valid perturbations that can plausibly serve as good global counterfactual explanations for cluster C_k.

The **effectiveness** (or **coverage**) of a perturbation set $\mathbb{S}$ represents the fraction of time-series in C_k for which at least one perturbation $\boldsymbol{\delta} \in \mathbb{S}$ results in a cluster flip:

$$\text{eff}(C_k, \mathbb{S}) = \frac{|\text{cov}(C_k, \mathbb{S})|}{|C_k|},$$

where the **covered population** $\text{cov}(C_k, \mathbb{S})$ consists of all $\boldsymbol{x} \in C_k$ such that $\exists \boldsymbol{\delta} \in \mathbb{S}$ where $\arg\max g(\boldsymbol{x} + \boldsymbol{\delta}) \neq \arg\max g(\boldsymbol{x})$. To quantify the effort required for this transition, we define the **flipping cost** as the minimum perturbation cost among all successful candidates in $\mathbb{S}$:

$$\text{cost}_{flip}(\boldsymbol{x}, \mathbb{S}) = \min\{\text{cost}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}) \mid \boldsymbol{\delta} \in \mathbb{S} \text{ s.t. a flip occurs}\}.$$

The **average flipping cost** serves as a global proxy for recourse effort, calculated as the mean cost over the covered population:

$$\text{afc}(C_k, \mathbb{S}) = \frac{\sum_{\boldsymbol{x} \in \text{cov}(C_k, \mathbb{S})} \text{cost}_{flip}(\boldsymbol{x}, \mathbb{S})}{|\text{cov}(C_k, \mathbb{S})|}$$

Standard multi-objective optimization of these criteria typically yields a Pareto front of non-dominated and incomparable solutions. For instance, increasing coverage typically necessitates a higher cost or a larger set of explanations [14, 20]. To resolve these competing trade-offs without resorting to arbitrary weight-tuning, we employ the *Minimum Description Length (MDL) principle* [31]. MDL treats the selection problem as a communication task, preferring the subset of perturbations that yields the shortest joint description of the model M and the observed data D. Formally, we seek:

$$M^{\star} \;=\; \arg\min_{M} L(D, M), \qquad L(D, M) \;=\; L(M) + L(D \mid M),$$

where $L(M)$ represents the complexity of the global explanation set, and $L(D \mid M)$ quantifies the "unexplained" data instances for which no selected perturbation provides a valid cluster flip.

In the MDL framework, each subset of perturbations $\mathbb{S} \subseteq \mathbb{\Delta}_k$ acts as a hypothesis to explain the transitions of instances in C_k. We define the **Model Code Length** $L(M)$ to penalize complexity across three dimensions: cardinality, structural sparsity, and magnitude (proximity), ensuring that the global explanation set is both effective and concise. The description length is formulated as:

$$\begin{aligned} L(M) &= L(\mathbb{S}) \\ &= \sum_{\boldsymbol{\delta} \in \mathbb{S}} \big(\text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1)\big) + (|\text{cov}(C_k, \mathbb{S})| + |\mathbb{S}|) \cdot p_sz \end{aligned}$$

The first term, $\text{bits}(\|\boldsymbol{\delta}\|_0)$, encodes the indices of the perturbed timesteps, where a universal code for integers penalizes non-sparse edits by requiring longer bit-lengths for complex structures. The second term, $\log_2(\|\boldsymbol{\delta}\|_1)$, captures the perturbation magnitude; following information-theoretic precision arguments, the bits required to represent a real-valued magnitude scale logarithmically, thereby favoring proximal edits. Finally, the assignment mapping term $(|\text{cov}(C_k, \mathbb{S})| + |\mathbb{S}|) \cdot p_sz$ accounts for the pointers required to associate each covered instance in C_k with its "best-fit" perturbation in $\mathbb{S}$. This structured approach transforms the multi-objective Pareto trade-off into a single communication task, where the most informative model is the one that minimizes the total description length of the cluster transitions.

The **Data Length** $L(D \mid M)$ quantifies the "failure to explain" by penalizing instances that remain uncovered by the selected set $\mathbb{S}$. In an information-theoretic sense, instances that do not change clusters under any $\boldsymbol{\delta} \in \mathbb{S}$ must be encoded as outliers. To avoid the prohibitive computational cost of calculating individual counterfactuals for every outlier, we approximate their encoding cost using a "worst-case" reference perturbation $\boldsymbol{\delta}^*$, defined as the most complex candidate in $\mathbb{\Delta}_k$. The data code length is formulated as:

$$\begin{aligned} L(D \mid M) &= L(C_k \mid \mathbb{S}) = \\ &= \big|C_k \setminus \text{cov}(C_k, \mathbb{S})\big| \cdot \Big(\text{bits}\big(\|\boldsymbol{\delta}^*\|_0\big) + \log_2(\|\boldsymbol{\delta}^*\|_1) + 2 \cdot p_sz\Big) \end{aligned}$$

This term acts as a dynamic penalty for low coverage: as the number of uncovered instances increases, the description length $L(D \mid M)$ grows linearly, forcing the MDL objective to favor the inclusion of more effective perturbations. By combining $L(M)$ and $L(D \mid M)$, the framework identifies the optimal $\mathbb{S}$, where the gain in coverage no longer justifies the added complexity, resolving the global selection problem without the need for manual hyperparameter tuning.

Problem 2 (Global Level Counterfactuals). *Given cluster C_k, a maximum cardinality of the perturbation set μ, find the optimal subset $\mathbb{S}^* \subseteq \mathbb{\Delta}_k$ that represents the solution of the following optimization problem:*

$$\mathbb{S}^* = \arg\min_{\mathbb{S} \subseteq \mathbb{\Delta}_k} \quad L(\mathbb{S}) + L(C_k \mid \mathbb{S}) \quad \textit{s.t.} \quad |\mathbb{S}| \leq \mu$$

4 GALACTIC Framework

4.1 Local Approach

Given a target cluster k', we seek a counterfactual $\boldsymbol{x}^{cf}$ that minimizes a cost function while traversing the decision boundary of the surrogate. To ensure structural relevance, we guide the search using a multi-stage segmentation and importance-scoring framework.

Segmentation and Subgrouping. We first capture the distinctive temporal properties of cluster C_k by computing instance-wise segmentations $\mathcal{S}(\boldsymbol{x})$ via change-point detection [34]. To account for imperfect clustering and internal variability, we encode these segmentations into gap vectors $\gamma(\boldsymbol{x})$ (representing segment durations) and apply K_{seg}-medoids clustering. This yields a set of representative patterns $\mathcal{P}_k = \{\mathcal{S}_1, \ldots, \mathcal{S}_R\}$ that summarizes the dominant temporal structures of the cluster. Each pattern $\mathcal{S}_r$ defines a subgroup G_r within cluster C_k, providing a structured partition of the cluster.

Importance-Guided Perturbation. For every cluster C_k, we derive **a timestep importance mask** $\boldsymbol{w} \in \{0, 1\}^T$ by evaluating the impact of segment permutations $\mathcal{P}$ on the accuracy of the surrogate model g. We define the importance of segment I_m as $\text{imp}(I_m) = |a_c - \frac{1}{B} \sum a_{c,m}^{(b)}|$, where a_c is the baseline accuracy for the cluster and $a_{c,m}^{(b)}$ is the accuracy after shuffling the values of segment I_m across the cluster population (in the b-th repetition. We propagate segment scores to timesteps and binarize using a quantile threshold (typically the 75th percentile). This results in a sparse mask $\boldsymbol{w}$ where $w_t = 1$ only for timesteps residing within highly discriminative segments. Similarly, we derive subgroup-specific importance vectors by restricting this permutation analysis to instances within the specific subgroup G_r and its corresponding representative pattern $\mathcal{S}_r$.

Rather than a penalty in the loss function, this mask $\boldsymbol{w}$ is applied as a hard constraint directly to the gradient during optimization: $\nabla \leftarrow \nabla_{\boldsymbol{x}'} \mathcal{L} \odot \boldsymbol{w}$. This projection ensures that the optimizer is restricted from altering "low-importance" regions, effectively preserving the non-discriminative parts of the time-series and forcing the

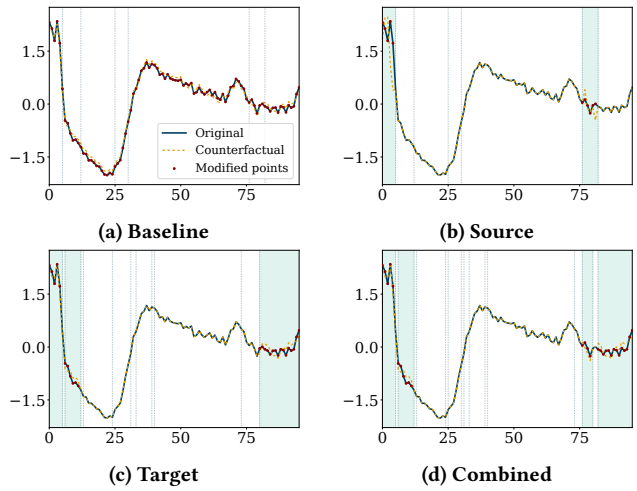


Figure 2: Comparison of local explanations for a time-series generated under different importance weighting strategies.

search to find a solution within the structural boundaries of the cluster. We consider three **strategies** for w for each $\boldsymbol{x}$: (1) Source-group importance, focuses on regions that define the current assignment (Figure 2b); (2) Target-cluster importance, identifies regions required to achieve membership in $C_{k'}$ (Figure 2c); and (3) Combined importance, which aligns breakpoints from both source group and target cluster to highlight mutually critical intervals (Figure 2d). Detailed formalizations are provided in the Appendix Section D.4.

Gradient-Based Optimization. We formalize the local counterfactual search as a constrained optimization problem, as detailed in Algorithm 1. We initialize the search by adding small Gaussian noise $\mathcal{N}(0, \epsilon^2)$ to the original series $\boldsymbol{x}$, restricted to the important regions defined by $\boldsymbol{w}$. The objective function $\mathcal{L}$ balances a proximity term $\mathcal{L}_{prox}$, ensuring the counterfactual remains close to the original instance, and a validity loss $\mathcal{L}_{flip}$ that drives the instance across the cluster boundary. Depending on the availability of a specific target cluster tk, $\mathcal{L}_{flip}$ either minimizes the negative log-likelihood of the target cluster $g_t(\boldsymbol{x}')$ or maximizes the distance from the distribution of the source cluster $g_s(\boldsymbol{x}')$.

At each iteration, we compute the gradient $\nabla_{\boldsymbol{x}'}\mathcal{L}$ and apply the structural importance mask via an element-wise product $\nabla_{\boldsymbol{x}'}\mathcal{L} \odot \boldsymbol{w}$. This projection ensures that the optimization updates are strictly confined to discriminative segments, preventing perturbations in non-essential regions. We utilize the Adam optimizer to update $\boldsymbol{x}'$ until the predicted label flips and the loss converges within a tolerance τ over p consecutive iterations. If multiple valid counterfactuals are found, the algorithm returns the candidate $\boldsymbol{x}^{cf}$ that minimizes the total cost, thereby satisfying the requirements for sparsity, proximity, and structural relevance.

4.2 Global Approach

The global counterfactual explanation problem requires selecting a small set of perturbations that induces cluster transitions for the largest possible fraction of the population. Since the space of all valid temporal perturbations $\mathbb{\Delta}$ is theoretically infinite and not

Algorithm 1 GALACTIC-L

Input: original series $\boldsymbol{x} \in \mathbb{R}^T$, surrogate g, step size η, iterations N_{iter}, weighting strategy $\mathcal{W}$, tolerance τ, patience p, jitter ϵ
Output: counterfactual $\boldsymbol{x}^{cf}$

procedure GALACTIC_L($\boldsymbol{x}, g, \eta, N_{iter}, \mathcal{W}, \tau, p, \epsilon$)
  $s \leftarrow \arg\max g(\boldsymbol{x})$ ▷ Source Cluster
  $t \leftarrow$ RESOLVETARGET$(\boldsymbol{x}, g)$ ▷ Target cluster or *None*
  $\boldsymbol{w} \leftarrow$ GETMASK$(\boldsymbol{x}, s, t, \mathcal{W})$ ▷ Derive Importance Mask
  $\boldsymbol{x}' \leftarrow \boldsymbol{x} + \mathcal{N}(0, \epsilon^2) \odot \boldsymbol{w}$ ▷ Stochastic initialization
  $\boldsymbol{x}^{cf} \leftarrow$ None, $\mathcal{L}_{min} \leftarrow \infty$, $\mathcal{L}_0 \leftarrow 0$, count $\leftarrow 0$
  for $i = 1$ to N_{iter} **do**
    $\mathcal{L}_{prox} \leftarrow \|\boldsymbol{x}' - \boldsymbol{x}\|_2$ ▷ Proximity loss
    if $t \neq$ None **then** ▷ Target-specific push
      $\mathcal{L}_{flip} \leftarrow -\log(g_t(\boldsymbol{x}') + \zeta)$ ▷ Validity loss
    else ▷ Agnostic push-away
      $\mathcal{L}_{flip} \leftarrow \log(g_s(\boldsymbol{x}') + \zeta)$
    end if
    $\mathcal{L}_i \leftarrow \lambda_1 \mathcal{L}_{prox} + \lambda_2 \mathcal{L}_{flip}$
    is_flipped $\leftarrow (t \neq \text{None} \wedge \arg\max g(\boldsymbol{x}') = t) \vee (t = \text{None} \wedge \arg\max g(\boldsymbol{x}') \neq s)$
    if is_flipped **then**
      if $\mathcal{L}_i < \mathcal{L}_{min}$ **then**
        $\boldsymbol{x}^{cf} \leftarrow \boldsymbol{x}'$, $\mathcal{L}_{min} \leftarrow \mathcal{L}_i$
      end if
      count $\leftarrow$ count + 1 if $(|\mathcal{L}_i - \mathcal{L}_{i-1}| < \tau)$ else 0
    end if
    if count $\geq p$ **then break**
    end if
    $\nabla \leftarrow \nabla_{\boldsymbol{x}'}\mathcal{L} \odot \boldsymbol{w}$ ▷ Apply structural mask to gradient
    $\boldsymbol{x}' \leftarrow$ AdamStep$(\boldsymbol{x}', \nabla, \eta)$ ▷ Update $\boldsymbol{x}'$
  end for
  return $\boldsymbol{x}^{cf}$
end procedure

explicitly defined, we must construct a pool of physically plausible and structurally relevant perturbations.

Candidate Generation. We leverage GALACTIC-L (Algorithm 1) to generate a finite candidate pool $\Delta_k = \{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_M\}$ by computing the perturbations $\boldsymbol{\delta}_i = \boldsymbol{x}_i^{cf} - \boldsymbol{x}_i$ obtained from a representative subset of instances in cluster C_k. This ensures that every candidate in our pool is a valid, gradient-verified transition that respects the cluster-specific structural constraints $\boldsymbol{w}$. By using these local "best-responses" as our search space, the global selection problem transforms into finding a subset $\mathbb{S} \subseteq \Delta_k$ that explains the transitions of the entire cluster through the shortest joint description length.

To ensure the candidate pool Δ_k is both computationally tractable and distributionally representative, we must carefully select the instances used for generation. Simple centroids are insufficient for time-series data as they often fail to capture the multi-modal nature of temporal clusters and may result in patterns that do not exist in the raw data. Instead, we select a distributional coreset of representative instances using the MMD-Critic algorithm [21].

MMD-Critic utilizes the Maximum Mean Discrepancy (MMD) to quantify the distance between the cluster distribution and the subset distribution in a Reproducing Kernel Hilbert Space (RKHS) [17]. This allows us to identify a set of prototypes, which represent the dominant temporal patterns of the cluster, and criticisms, which represent low-density regions and outliers that prototypes fail to

capture. By generating perturbations from both prototypes and criticisms, we ensure the candidate pool Δ_k contains transitions for both the "average" and the "edge cases" of the cluster manifold, providing a robust foundation for the subsequent MDL selection.

Algorithmic Variants. While the MDL objective can evaluate any subset size, interpretability requirements dictate a limit of at most μ_k global perturbations per cluster. We present four variants of the selection process. A detailed comparative evaluation of the quality-runtime Pareto frontier for these is provided in the Section F.2.

1. Optimal. The Optimal approach (App. Algorithm 3) performs an exhaustive search over the power set of candidates of cardinality at most μ. It evaluates $\sum_{i=1}^{\mu} \binom{|\Delta_k|}{i}$ combinations to find the global minimum of the joint description length $L(C_k, \mathbb{S})$. While computationally prohibitive for large sets, due to its complexity ($O(|\Delta_k|^{\mu_k})$), it serves as the ground-truth baseline for small-scale patterns.

2. Greedy. The Greedy approach (App. Algorithm 4) provides an efficient approximation of the optimal solution by iteratively incorporating the perturbation that yields the maximal marginal gain in compression. Under mild assumptions (see Section B), the MDL objective behaves as a supermodular function, rendering the reduction in total description length a monotone submodular set function. This property ensures that the greedy procedure achieves the $(1 - 1/e)$ approximation guarantee for submodular maximization [18] with a computational complexity of $O(\mu_k|\Delta_k|)$. The search terminates when the budget μ_k is reached or an intrinsic stopping criterion is met: if the marginal bit-cost of encoding a new perturbation exceeds its coverage benefit, the algorithm stops to preserve model parsimony and avoid overfitting.

3. Hierarchical. The Hierarchical approach (Algorithm 2) decomposes the selection into two phases: subgroup level selection followed by cluster-level refinement. This architecture supports two distinct operational modes: **(a) Hierarchical Optimal.** It employs exhaustive search at the subgroup level to identify the exact MDL-minimizing perturbations for each $G \subset C_k$.With m_G candidates per group and a subgroup budget μ_r, Phase 1 requires $\sum_G \sum_{s=1}^{\mu_r} \binom{m_G}{s}$ evaluations. The resulting winners form a reduced set $\widetilde{\Delta}_k$ of size m', which is then refined via a second exhaustive search requiring $\sum_{s=1}^{\mu_k} \binom{m'}{s}$ evaluations. This significantly reduces the search space compared to a flat optimal search, though it sacrifices global optimality for local precision. **(b) Hierarchical Greedy.** This approach is designed for maximum scalability and utilizes the submodular greedy strategy at both levels. At the subgroup level, it requires $\sum_G \mu_r m_G$ evaluations to populate the reduced set $\widetilde{\Delta}_k$. The final cluster-level refinement adds $\mu_k m'$ evaluations. This variant drastically reduces the number of MDL assessments, especially when $m' \ll |\Delta_k|$, making it the most suitable approach for high-dimensional time series and massive datasets.

Algorithm 2 Galactic-G: Hierarchical

Input: Cluster C_k, groups $\mathcal{G}_k$, candidate perturbations Δ_{C_k}, budget μ_k, strategy algo $\in$ {`optimal`, `greedy`}
Output: Global perturbation set $\mathbb{S}_{C_k}$

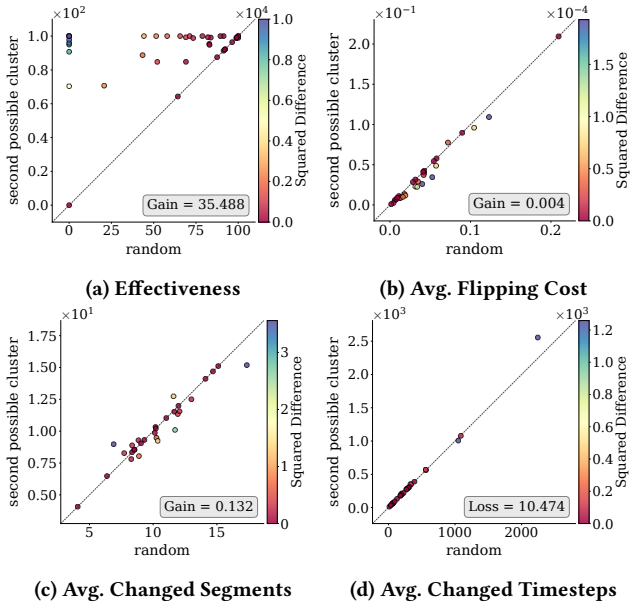
```
procedure HIERARCHICAL(C_k, G_k, Δ_{C_k}, μ_k, algo):
    Δ̃_{C_k} ← ∅ ▷ Pool of candidate perturbations from group selections
    // Phase 1: Subgroup level selection
    for all G_r ∈ G_k do
        Δ_{G_r} ← GETPERM(Δ_{C_k}, G_r, g)          ▷ Get applicable candidate
        μ_r ← ⌈ (|G_r| / |C_k|) μ_k ⌉ · |G_k|        ▷ Proportional budget allocation
        Δ̃_{C_k} ← Δ̃_{C_k} ∪ SELECTPERM(Δ_{G_r}, G_r, μ_r, algo)  ▷ Group MDL
    end for
    // Phase 2: Cluster level refinement
    S_{C_k} ← SELECTPERM(Δ̃_{C_k}, C_k, μ_k, algo)     ▷ Refined cluster MDL
    return S_{C_k}
end procedure

procedure SELECTPERM(Δ, C, μ, algo):
    if algo = optimal then
        return OPTIMAL(Δ, C, μ)      ▷ Exhaustive search for small |Δ|
    else if algo = greedy then
        return GREEDY (Δ, C, μ)                      ▷ Greedy MDL
    end if
end procedure
```

5 Experimental Evaluation

To evaluate Galactic, we conduct experiments on diverse datasets from the UCR Time-Series Archive [10] addressing four research directions: (i) **Target Selection Analysis**: Validating the superiority of the *second possible cluster* policy over *random* selection; (ii) **Weighting Strategy Efficacy**: Demonstrating the performance of *Combined* segment importance; (iii) **Local Evaluation**: Benchmarking Galactic-L against state-of-the-art local counterfactual generators; (iv) **Global Evaluation**: Benchmarking Galactic-G approaches against adapted baselines for clustered series.

Datasets and Models. We evaluate Galactic on 30 datasets from the UCR Time Series Archive [10], spanning multiple domains (e.g., Image, Motion, Sensor). Although the UCR archive is primarily used for classification benchmarks, it is also used for clustering [13, 29]. We treat the ground-truth labels of each UCR dataset as the cluster partition (i.e., each class defines a cluster C_k), while subgroups are obtained by clustering the segmentation gap vectors within each cluster as described in Section 4.1. Appendix Table 4 summarizes the selected datasets, including their size, length, number of clusters, domain type, and the classification accuracy achieved by the surrogate model. All surrogate models follow an 80% to 20% train-test split. Detailed parameter selection, architectural specifications, hardware configurations, and datasets are described in Section D. The source code of Galactic is available upon request.

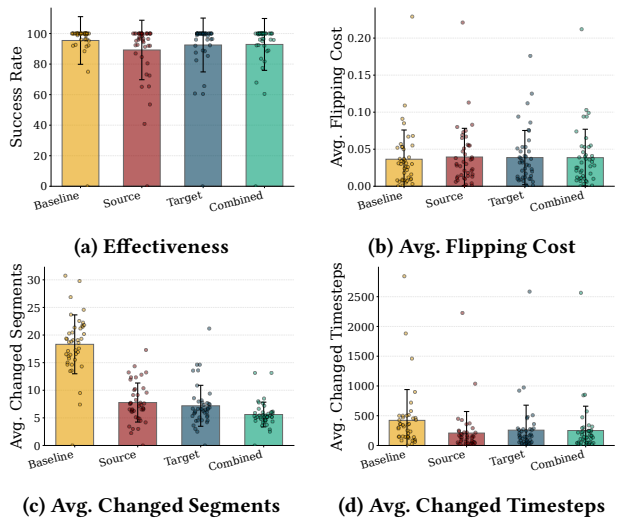
Evaluation Metrics. To assess the performance of the generated counterfactuals, we report five standard measurements across all experiments: (1) **Effectiveness (eff)**: the percentage of instances where the search successfully flips the prediction to the target cluster; (2) **Average Flipping Cost (afc)**: the L_2 distance between the original series x and the counterfactual x^{cf}, representing the minimality of the edit; (3) **Average Changed Segments (acs)**: the average number of segments in the series that were modified, i.e., the *localized* changes within contiguous regions rather than scattered ones across disjoint timesteps; (4) **Average Changed Timesteps (act)**: the average number of individual timesteps modified, reflecting fine-grained interpretability; and (5) **Runtime (RT)**: the total time required to generate explanations.



(a) Effectiveness (b) Avg. Flipping Cost

(c) Avg. Changed Segments (d) Avg. Changed Timesteps

Figure 3: Galactic-L Target Selection Analysis. Comparison of second possible and random policies across UCR datasets.



(a) Effectiveness (b) Avg. Flipping Cost

(c) Avg. Changed Segments (d) Avg. Changed Timesteps

Figure 4: Importance Weighting Strategies Comparison. Points denote individual datasets; bars indicate dataset-wide averages and standard deviations for validity and sparsity.

5.1 Search Policy: Target Selection

For Galactic-L, we perform a comparative evaluation of two target selection policies: **second possible** (selecting the cluster with the next-highest surrogate probability), and **random** (randomly selecting a target cluster). To compare policies across datasets, we summarize their behavior using the aggregate measurement *Gain/Loss* that reflects whether a policy improves or worsens a metric on average relative to another policy (a Gain indicates a higher success rate for the method reported in the y-axis, lower costs (flipping cost, changed segments/timesteps, and runtime)). Details can be found in the Section D.3, as well as the evaluation of a third selection policy **all_random** (no explicit target selection) in the Section F.1. As shown in Figure 3, the *second possible* policy consistently achieves the highest effectiveness while minimizing average flipping cost. In contrast, *random* exhibit higher average flipping cost distances, a higher number of changed segments, and lower effectiveness, validating that the Galactic objective effectively leverages the latent topology of the classifier to find efficient transition paths.

5.2 Importance Weighting: Sparsity and Validity

We benchmark our three weighting strategies $\mathcal{W}$ to assess their impact on counterfactual quality against the **Baseline** (no masking): **Source**, **Target**, and **Combined**. As shown in Figure 4, the Combined strategy yields the optimal trade-off between cluster-flip success and interpretability. While the Baseline often achieves high success rates, it suffers from poor sparsity, modifying irrelevant temporal regions. Conversely, the Combined mask restricts perturbations to segments critical to both source and target clusters, significantly reducing the number of changed segments (as evidenced by the lower average and narrower standard deviation in sparsity metrics) without compromising the cost minimality of the resulting counterfactual.

5.3 Local Evaluation

We benchmark the Galactic-L against three state-of-the-art baselines that represent different paradigms of counterfactual search: k-Nearest Neighbors (k-NN) (retrieval), TSEvo [19] (evolutionary optimization), and Glacier [40] (gradient-based constrained search), evaluating both its Glacier-Local and Glacier-Global variants. Implementation details and method-specific configurations are reported in Section E. Table 1 presents the summarized results of all competing methods across 10 UCR datasets; results for an additional 20 datasets are provided in Appendix Table 5.

The retrieval-based k-NN baseline serves as an informative lower bound: although it achieves high success rates when nearby cross-cluster instances exist, it consistently incurs high perturbation cost and poor sparsity, as retrieved neighbors often differ from the query across large portions of the series, with effectiveness degrading in sparse regions of the data. TSEvo attains high validity due to its exploratory black-box evolutionary search, but at the expense of substantially higher flipping costs, extensive multi-segment edits, and orders-of-magnitude higher runtimes, yielding less localized and harder to interpret explanations. Glacier improves upon black-box methods in both cost and efficiency through gradient-based, importance-aware constraints; however, its reliance on soft penalties allows perturbations to leak into non-critical regions, often resulting in more modified segments and timesteps than necessary, particularly for the global variant, where a single mask is applied across heterogeneous cluster members.

In contrast, Galactic-L achieves the most favorable trade-off across all dimensions, exceeding the effectiveness of gradient-based

Table 1: Local Performance Benchmarking. Comparative analysis of Galactic-L against kNN, TSEvo, and Glacier variants across effectiveness, average flipping cost, average segments changed, average timesteps changed, and runtime. The null indicator "-" is assigned throughout the metrics where the corresponding effectiveness is zero. Results for additional datasets can be found in Appendix Table 5.

Dataset	KNN					TSEvo					Glacier-L					Glacier-G					Galactic-L				
	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT
ACSF1	100	0.08	24.82	1460	0.15	100	0.09	24.82	1460	405.54	28.33	0.01	21.88	1460	9.69	26.67	0.01	21.94	1459.94	9.95	98.33	0.03	7.75	892.08	22.12
ArrowHead	100	0.1	35.79	251	0.17	100	0.09	35.79	251	385.22	36.51	0	34.22	251	11.06	52.38	0.01	33.73	251	18.27	98.41	0.02	9.47	131.39	11.21
Car	100	0.12	27.61	577	0.07	100	0.17	27.61	577	240.77	69.44	0.01	27.44	577	12.04	88.89	0.01	27.25	576.97	15.22	100	0.03	7.36	318.06	5.13
CBF	100	0.16	26.49	128	1.34	100	0.19	26.49	128	1751.15	0.36	0.02	28	128	0.64	0.36	0.02	28	128	0.63	76.7	0.09	8.55	53.03	158.08
Coffee	100	0.24	14.76	286	0.05	100	0.21	14.76	286	105.57	29.41	0.06	15.6	286	3.14	58.82	0.06	15.3	286	5.99	94.12	0.18	4	119.75	4.82
ECG200	0	-	-	-	0.15	100	0.12	19.38	96	340.19	33.33	0.01	20	96	10.95	45	0.01	20.07	96	23.83	95	0.04	5.25	47.11	32.13
Fish	100	0.1	29.45	463	0.28	100	0.13	29.45	463	1071.66	30.48	0.01	29.22	462.94	25.33	40	0.01	29.5	463	29.91	99.05	0.04	7.85	258.37	23.2
SyntheticControl	100	0.21	14.71	60	0.27	100	0.23	14.71	60	570.44	1.11	0.02	17	60	1.32	0	-	-	-	14.91	91.67	0.1	5.01	24.73	199.21
UMD	0	-	-	-	0.18	100	0.18	25.59	149.8	317.75	18.52	0.02	27.6	150	5.02	25.93	0.03	27.43	150	12.36	98.15	0.11	7.04	80.92	30
Worms	100	0.18	37.24	900	0.43	100	0.15	37.24	900	499.46	36.76	0	36.56	900	14.59	41.18	0	36.39	899.93	14.9	98.53	0.01	9.79	464.4	8.67

Table 2: Global Performance Benchmarking. Comparative analysis of Galactic-G variants against GLOBE-CE* and Glacier-G* across effectiveness, average flipping cost, average segments changed, average timesteps changed, and runtime. The * indicates adaptations for the global timeseries clustering context; the null indicator "-" is assigned throughout the metrics where the corresponding effectiveness is zero. Results for additional datasets can be found in Appendix Table 6.

Dataset	Galactic-G																				Competitors									
	Optimal					Greedy					Hierarchical					Hierarchical Greedy					GLOBE-CE*					Glacier-G*				
	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT
ACSF1	62.8	0.031	3.7	813	4.17	62.8	0.021	3.7	812	0.66	37.6	0.010	3.4	591	1.49	37.6	0.009	3.2	581	0.89	21.2	0.006	4.0	730	60.93	31.2	0.006	5.4	1022	0.01
ArrowHead	75.9	0.021	5.0	102	49.55	75.9	0.019	4.7	102	5.34	72.0	0.021	4.9	97	3.34	72.0	0.019	4.6	102	2.80	54.6	0.088	9.4	251	35.57	20.2	0.006	7.9	167	10.30
Car	70.8	0.024	4.8	350	106.68	70.8	0.023	4.8	361	3.18	55.2	0.015	4.4	282	3.24	58.6	0.015	4.2	280	2.41	74.2	0.056	6.7	433	32.16	33.4	0.008	8.8	577	0.00
CBF	39.4	0.105	3.3	47	36.93	39.4	0.105	3.3	47	10.21	33.7	0.098	3.3	51	1.57	33.7	0.094	3.3	51	1.65	26.0	0.186	7.1	128	50.57	0.0	-	-	-	0.00
Coffee	100.0	0.186	4.3	107	3.32	100.0	0.177	4.4	114	2.15	100.0	0.186	4.3	107	1.59	100.0	0.177	4.4	114	1.59	43.1	0.436	4.9	143	26.52	52.4	0.074	9.5	286	7.25
ECG200	58.8	0.056	4.0	42	287.92	58.8	0.056	4.0	42	17.93	53.5	0.048	4.0	40	12.11	57.6	0.053	4.0	41	8.48	59.8	0.272	8.4	96	22.13	6.2	0.018	13.2	96	25.75
Fish	87.4	0.040	5.5	286	5929.39	87.4	0.034	5.2	245	18.64	67.6	0.035	5.8	268	8.80	68.8	0.033	5.3	258	4.77	82.5	0.109	7.2	397	49.87	31.1	0.011	9.5	463	27.03
SyntheticControl	72.3	0.115	3.3	19	136.92	72.3	0.106	3.3	19	23.68	69.5	0.110	3.5	19	20.21	69.7	0.107	3.6	20	18.03	68.0	0.280	6.5	50	29.96	0.0	-	-	-	14.91
UMD	84.2	0.104	6.4	75	1491.24	84.2	0.093	6.0	70	18.89	81.9	0.102	6.6	80	14.02	81.9	0.091	6.1	72	9.72	66.1	0.059	9.1	100	37.35	14.4	0.020	11.5	100	19.34
Worms	61.0	0.012	7.5	427	213.45	61.4	0.012	8.0	437	5.17	51.9	0.011	8.1	447	10.94	54.0	0.011	8.1	437	8.51	35.1	0.019	16.6	900	57.71	20.4	0.006	21.0	900	0.00

baselines while producing substantially sparser, low-cost, and cluster-specific counterfactuals. By restricting optimization to transition-critical temporal regions per cluster, Galactic-L preserves the global structure of the series and offers explanations that are both highly interpretable and computationally efficient, being orders of magnitude faster than evolutionary methods and consistently faster than gradient-based approaches.

5.4 Global Evaluation

This experiment evaluates the quality and computational efficiency of the global selection phase of our four Galactic-G variants (Optimal, Greedy, and their respective Hierarchical extensions) against two state-of-the-art global counterfactual baselines adapted for the clustering temporal domain. In the absence of native global generators for time-series clustering, we evaluate against Glacier-G*, an adaptation of the Glacier framework [40], where local perturbations are aggregated from all correctly clustered instances via k-means clustering to extract representative centroids, and GLOBE-CE*, an adaptation of GLOBE-CE [23], that flattens sequences into high-dimensional vectors to optimize global directions through random sampling, and uses bisection-based scalar optimization to maximize effectiveness. Table 2 presents summarized results for all competing methods across 10 UCR datasets; results for an additional 20 datasets are provided in Appendix Table 6.

GLOBE-CE* relies on a tabularized view of the series that ignores temporal dependencies, resulting in global deltas that lack structural sparsity and generalize poorly. Its optimization strategy further amplifies this limitation: it generates large sets of random translation vectors and applies bidirectional scaling to increase coverage, which introduces substantial computational overhead compared to Galactic-G variants and Glacier-G*. Since this process is not guided by the temporal structure or the cluster manifold, it often requires large-magnitude shifts to capture outlying instances, ultimately leading to inflated average flipping costs.

The Glacier-G* approach struggles to provide adequate effectiveness at the cluster level. Aggregating individually generated deltas does not sufficiently capture the underlying cluster structure, resulting in global directions that remain instance-specific and fail to represent coherent, cluster-level transformations.

Galactic-G consistently outperforms adapted baselines, achieving a superior effectiveness-to-cost trade-off with significantly lower computational latency. The MDL-based selection identifies the most representative and "compressible" perturbations. This strategy respects the internal segmentation of the cluster, resulting in global explanations that maximize population coverage, while minimizing both the number of modified segments and total flipping cost. Consequently, Galactic-G provides summaries that are both more interpretable and computationally efficient than traditional aggregation or random-search methods.

6 Conclusions

In this work, we present Galactic, a unified framework bridging the gap between local and global explainability in time-series clustering. By rigorously formulating the explanation selection

as an MDL-based submodular maximization problem, we provide a theoretically grounded approach to summarize cluster transitions. Extensive empirical evaluation demonstrates that Galactic consistently outperforms state-of-the-art baselines in sparsity and coherence while maintaining superior computational efficiency.

Future work will extend this formalism to multivariate settings and integrate causal discovery to disentangle latent temporal confounders. Furthermore, we intend to explore how the MDL-based adaptation can be generalized to the tabular data domain for broader explainability applications.

7 Acknowledgments

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program, Horizon projects TwinODIS (101160009) and DataGEMS (101188416), $Y\Pi AI\Theta A$ & NextGenerationEU project HARSH ($Y\Pi 3TA - 0560901$) that is carried out within the framework of the National Recovery and Resilience Plan "Greece 2.0" with funding from the European Union - NextGenerationEU.

References

[1] Carlos Alzate and Johan A.K. Suykens. 2010. Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 2 (2010), 335–347. doi:10.1109/TPAMI.2008.292

[2] Emre Ates, Burak Aksar, Vitus J Leung, and Ayse K Coskun. 2021. Counterfactual explanations for multivariate time series. In *2021 international conference on applied artificial intelligence (ICAPAI)*. IEEE.

[3] Omar Bahri, Peiyu Li, Soukaina Filali Boubrahimi, and Shah Muhammad Hamdi. 2022. Temporal rule-based counterfactual explanations for multivariate time series. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*.

[4] Mohini Bariya, Alexandra von Meier, John Paparrizos, and Michael J Franklin. 2021. k-shapestream: Probabilistic streaming clustering for electric grid events. In *2021 IEEE Madrid PowerTech*.

[5] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. 2003. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[6] Jürgen Bernard, Tobias Ruppert, Maximilian Scherer, Tobias Schreck, and Jörn Kohlhammer. 2012. Guided discovery of interesting relationships between time series clusters and metadata properties. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*.

[7] Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. 2021. Interpretable clustering: an optimization approach. *Machine Learning* (2021).

[8] Angela Bonifati, Francesco Del Buono, Francesco Guerra, and Donato Tiano. 2022. Time2Feat: Learning interpretable representations for multivariate time series clustering. *Proceedings of the VLDB Endowment (PVLDB)* (2022).

[9] Paul Boniol, Donato Tiano, Angela Bonifati, and Themis Palpanas. 2025. k-Graph: a graph embedding for interpretable time series clustering. *IEEE Transactions on Knowledge and Data Engineering* (2025).

[10] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[11] Eoin Delaney, Derek Greene, and Mark T Keane. 2021. Instance-based counterfactual explanations for time series classification. In *International conference on case-based reasoning*.

[12] Doulaye Dembele and Philippe Kastner. 2003. Fuzzy C-means method for clustering microarray data. *bioinformatics* (2003).

[13] Dimitris Fotakis, Panagiotis Patsilinakos, Eleni Psaroudaki, and Michalis Xefteris. 2024. Efficient time-series clustering through sparse gaussian modeling. *Algorithms* 17, 2 (2024), 61.

[14] Christos Fragkathoulas, Vasiliki Papanikou, Evaggelia Pitoura, and Evimaria Terzi. 2025. FACEGroup: Feasible and Actionable Counterfactual Explanations for Group Fairness. In *ECML-PKDD*.

[15] Tak-chung Fu, Fu-lai Chung, Vincent Ng, and Robert Luk. 2001. Pattern discovery from stock time series using self-organizing maps. In *Workshop notes of KDD2001 workshop on temporal data mining*.

[16] Cyril Goutte, Peter Toft, Egill Rostrup, Finn Å Nielsen, and Lars Kai Hansen. 1999. On clustering fMRI time series. *NeuroImage* (1999).

[17] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* (2012).

[18] Dorit S Hochba. 1997. Approximation algorithms for NP-hard problems. *ACM Sigact News* (1997).

[19] Höllig et al. 2022. Tsevo: Evolutionary counterfactual explanations for time series classification. In *(ICMLA)*.

[20] Loukas Kavouras, Eleni Psaroudaki, Konstantinos Tsopelas, Dimitrios Rontogiannis, Nikolaos Theologitis, Dimitris Sacharidis, Giorgos Giannopoulos, Dimitrios Tomaras, Kleopatra Markou, Dimitrios Gunopulos, Dimitris Fotakis, and Ioannis Emiris. 2025. GLANCE: Global Actions in a Nutshell for Counterfactual Explainability. arXiv:2405.18921 [cs.LG] https://arxiv.org/abs/2405.18921

[21] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems* (2016).

[22] Jana Lang, Martin A Giese, Winfried Ilg, and Sebastian Otte. 2023. Generating sparse counterfactual explanations for multivariate time series. In *International Conference on Artificial Neural Networks*.

[23] Dan Ley, Saumitra Mishra, and Daniele Magazzeni. 2023. GLOBE-CE: A translation based approach for global counterfactual explanations. In *International conference on machine learning*. PMLR, 19315–19342.

[24] Laura Manduchi, Matthias Hüser, Martin Faltys, Julia Vogt, Gunnar Rätsch, and Vincent Fortuin. 2021. T-dpsom: An interpretable clustering method for unsupervised learning of patient health states. In *Proceedings of the Conference on Health, Inference, and Learning*.

[25] Vit Niennattrakul and Chotirat Ann Ratanamahatana. 2007. On clustering multimedia time series data using k-means and dynamic time warping. In *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*.

[26] Mandani Ntekouli, Gerasimos Spanakis, Lourens Waldorp, and Anne Roefs. 2024. Explaining Clustering of Ecological Momentary Assessment Data Through Temporal and Feature Attention. In *XAI*.

[27] Tim Oates, Laura Firoiu, and Paul R Cohen. 1999. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 workshop on neural, symbolic and reinforcement learning methods for sequence learning*.

[28] Ozan Ozyegen, Nicholas Prayogo, Mucahit Cevik, and Ayse Basar. 2022. Interpretable time series clustering using local explanations. *arXiv preprint arXiv:2208.01152* (2022).

[29] John Paparrizos, Fan Yang, and Haojun Li. 2024. Bridging the gap: A decade review of time-series clustering methods. *arXiv preprint arXiv:2412.20582* (2024).

[30] Mario Refoyo and David Luengo. 2024. Sub-space: Subsequence-based sparse counterfactual explanations for time series classification problems. In *XAI*.

[31] Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica* (1978).

[32] Advait Sarkar, Martin Spott, Alan F Blackwell, and Mateja Jamnik. 2016. Visual discovery and model-driven explanation of time series patterns. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*.

[33] Udo Schlegel, Gabriel Marques Tavares, and Thomas Seidl. 2025. Towards explainable deep clustering for time series data. *arXiv preprint arXiv:2507.20840* (2025).

[34] Torty Sivill and Peter Flach. 2022. Limesegment: Meaningful, realistic time series explanations. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

[35] Aurora Spagnol, Kacper Sokol, Pietro Barbiero, Marc Langheinrich, and Martin Gjoreski. 2024. Counterfactual Explanations for Clustering Models. *arXiv preprint arXiv:2409.12632* (2024).

[36] Dat Tran and Michael Wagner. 2002. Fuzzy c-means clustering-based speaker verification. In *AFSS International Conference on Fuzzy Systems*.

[37] Georgios Vardakas, Antonia Karra, Evaggelia Pitoura, and Aristidis Likas. 2025. Counterfactual Explanations for k-means and Gaussian Clustering. *arXiv preprint arXiv:2501.10234* (2025).

[38] Ulrike von Luxburg. 2007. A Tutorial on Spectral Clustering. *Statistics and Computing* 17, 4 (2007), 395–416. doi:10.1007/s11222-007-9033-z

[39] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* (2017).

[40] Zhendong Wang, Isak Samsten, Ioanna Miliou, Rami Mochaourab, and Panagiotis Papapetrou. 2024. Glacier: guided locally constrained counterfactual explanations for time series classification. *Machine Learning* (2024).

[41] Akihiro Yamaguchi, Ken Ueno, Ryusei Shingaki, and Hisashi Kashima. 2024. Learning counterfactual explanations with intervals for time-series classification. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*.

A Notation

A summary of the notation used throughout the paper is provided in Table 3.

Table 3: Notation Table for the Galactic Framework

Symbol	Meaning
$\mathcal{T}, T$	Space of series $\mathbb{R}^T$; Series length
$\boldsymbol{x} = [x_1, \ldots, x_T]$	A time-series instance of length T
$X = \{\boldsymbol{x}^{(i)}\}_{i=1}^N$	Dataset containing N series
$f : \mathcal{T} \to \{1, \ldots, K\}$	Underlying clustering algorithm
$C = \{C_1, \ldots, C_K\}$	Cluster partition
$g : \mathcal{T} \to [0,1]^K$	Probabilistic surrogate classifier
$\mathcal{S}(x)$	Set of M contiguous, disjoint temporal segments
x^{cf}, δ	Local counterfactual $x^{cf} = x + \delta$; Perturbation δ
$\text{cost}(\cdot)$	Perturbation cost metric (e.g., ℓ_p distance)
Δ_k	Candidate perturbations for cluster C_k
p_sz	The bit-cost of a pointer
M	The explanation model
D	The data population to be explained
$L(M) := L(\mathbb{S})$	Model description length
$L(D \mid M) := L(C_k \mid \mathbb{S})$	Data description length
$L(D, M) := L(C_k, \mathbb{S})$	MDL objective: $L(M) + L(D \mid M)$
$\text{cov}(C_k, \mathbb{S})$	Set of instances in C_k successfully flipped by $\mathbb{S}$
$\text{eff}(C_k, \mathbb{S})$	Effectiveness of the summary $\mathbb{S}$
$\text{cost}_{\text{flip}}(\boldsymbol{x}, \mathbb{S})$	Minimum cost to flip instance x using the set $\mathbb{S}$
$\text{afc}(C_k, \mathbb{S})$	Average Flipping Cost
$\Delta(\mathbb{S})$	MDL reduction: $L(D, \varnothing) - L(D, \mathbb{S})$

B Theoretical Foundations of Global Selection

The global selection phase of Galactic aims to identify a non-redundant set of representative perturbations that effectively summarize cluster transitions. As delineated in Section 2, this is framed as a Minimum Description Length (MDL) optimization problem. Given that the exhaustive search over the power set of all candidate perturbations is NP-hard, we employ a greedy approximation algorithm. To guarantee the theoretical bounds of this greedy approach, we provide here the formal proof that the MDL objective function exhibits *supermodularity*, and that the MDL reduction is *submodular*. For these proofs, we make the following assumption.

Assumption 1 (Admissible Additions). *When analyzing the addition of a perturbation $\boldsymbol{\delta} \in \Delta_k$, to our model $\boldsymbol{\delta} \notin \mathbb{S}$, we only allow admissible additions that newly cover at least one previously uncovered instance, i.e.,* $\text{cov}(C_k, \mathbb{S} \cup \{\boldsymbol{\delta}\}) \setminus \text{cov}(C_k, \mathbb{S}) \neq \emptyset$.

Proposition 1 (Supermodularity of the MDL objective). *For a source cluster C_k and the finite set of candidate perturbations Δ_k for C_k, the set function $L(C_k, \mathbb{S}) = L(\mathbb{S}) + L(C_k \mid \mathbb{S})$ is* supermodular *on Δ_k. That is, for all $\mathbb{A} \subseteq \mathbb{B} \subseteq \Delta_k$ and all $\boldsymbol{\delta} \in \Delta_k \setminus \mathbb{B}$, under Assumption 1:*

$$L(D, \mathbb{A} \cup \{\boldsymbol{\delta}\}) - L(D, \mathbb{A}) \;\leq\; L(D, \mathbb{B} \cup \{\boldsymbol{\delta}\}) - L(D, \mathbb{B}).$$

Proof. We analyze the marginal change in the total description length upon adding a new perturbation $\boldsymbol{\delta} \notin \mathbb{S}$ to an existing summary set $\mathbb{S}$. We define the uncovered set $\mathcal{R}(C_k, \mathbb{S}) = C_k \setminus \text{cov}(C_k, \mathbb{S})$.. The reference perturbation is the maximal-description perturbation

$$\boldsymbol{\delta}^* = \arg\max_{\boldsymbol{\delta} \in \Delta_k} \Big(\text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1) + 2p_sz \Big),$$

fixed and independent of $\mathbb{S}$. We will refer to the cost of any uncovered series as MC, where:

$$MC = \text{bits}(\|\boldsymbol{\delta}^*\|_0) + \log_2(\|\boldsymbol{\delta}^*\|_1) + 2p_sz.$$

Then $MC > 0$ and in particular $MC \geq 2p_sz > p_sz$.

When analyzing the addition of a perturbation to our model $\boldsymbol{\delta} \notin \mathbb{S}$, we only allow admissible additions that newly cover at least one previously uncovered instance, i.e., $\text{cov}(C_k, \mathbb{S} \cup \{\boldsymbol{\delta}\}) \setminus \text{cov}(C_k, \mathbb{S}) \neq \emptyset$. The marginal coverage gain provided by adding $\boldsymbol{\delta}$ to $\mathbb{S}$ is the magnitude of the set of instances covered by $\boldsymbol{\delta}$ that were not previously covered by $\mathbb{S}$:

$$\begin{aligned} \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) &= |\text{cov}(C_k, \mathbb{S} \cup \{\boldsymbol{\delta}\}) \setminus \text{cov}(C_k, \mathbb{S})| \\ &= |\text{cov}(C_k, \{\boldsymbol{\delta}\}) \cap \mathcal{R}(C_k, \mathbb{S})| \\ &= |\text{cov}(C_k, \mathbb{S} \cup \{\boldsymbol{\delta}\})| - |\text{cov}(C_k, \mathbb{S})| \end{aligned} \tag{1}$$

Since $\mathbb{A} \subseteq \mathbb{B}$ implies $\text{cov}(C_k, \mathbb{A}) \subseteq \text{cov}(C_k, \mathbb{B})$, we have $\mathcal{R}(C_k, \mathbb{B}) \subseteq \mathcal{R}(C_k, \mathbb{A})$ and thus

$$\begin{aligned} (\text{cov}(C_k, \{\boldsymbol{\delta}\}) \cap \mathcal{R}(C_k, \mathbb{B})) \subseteq (\text{cov}(C_k, \{\boldsymbol{\delta}\}) \cap \mathcal{R}(C_k, \mathbb{A})) \\ \implies \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{B}) \leq \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{A}). \end{aligned} \tag{2}$$

The marginal gain of the description length $L(C_k, \mathbb{S})$ can be decomposed into the model part and data part.

Model part. The marginal increment of the model part is:

$$\Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{S}) = L\big(\mathbb{S} \cup \{\boldsymbol{\delta}\}\big) - L\big(\mathbb{S}\big),$$

where

$$\begin{aligned} L\big(\mathbb{S} \cup \{\boldsymbol{\delta}\}\big) = &\sum_{\boldsymbol{\delta}' \in \mathbb{S} \cup \{\boldsymbol{\delta}\}} \Big(\text{bits}(\|\boldsymbol{\delta}'\|_0) + \log_2(\|\boldsymbol{\delta}'\|_1) \Big) \\ &+ \big(|\text{cov}(C_k, \mathbb{S} \cup \{\boldsymbol{\delta}\})| + |\mathbb{S} \cup \{\boldsymbol{\delta}\}| \big)\, p_sz, \end{aligned} \tag{3}$$

$$\begin{aligned} L\big(\mathbb{S}\big) = &\sum_{\boldsymbol{\delta}' \in \mathbb{S}} \Big(\text{bits}(\|\boldsymbol{\delta}'\|_0) + \log_2(\|\boldsymbol{\delta}'\|_1) \Big) \\ &+ \big(|\text{cov}(C_k, \mathbb{S})| + |\mathbb{S}| \big)\, p_sz. \end{aligned} \tag{4}$$

Hence,

$$\begin{aligned} \Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{S}) \overset{(3),(4)}{=} &\Big(\text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1) \Big) \\ &+ \Big(|\text{cov}(C_k, \mathbb{S} \cup \{\boldsymbol{\delta}\})| - |\text{cov}(C_k, \mathbb{S})| \Big) p_sz \\ &+ \Big(|\mathbb{S} \cup \{\boldsymbol{\delta}\}| - |\mathbb{S}| \Big) p_sz. \overset{(1)}{\implies} \\ \Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{S}) = &\text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1) \\ &+ \big(\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) + 1 \big) p_sz \end{aligned} \tag{5}$$

Data part. As defined in Section 3,

$$L(C_k \mid \mathbb{S}) = |C_k \setminus \text{cov}(C_k, \mathbb{S})| \cdot MC = |\mathcal{R}(C_k, \mathbb{S})| \cdot MC.$$

When adding $\boldsymbol{\delta}$ to $\mathbb{S}$, the uncovered set shrinks by exactly the number of newly covered instances:

$$|\mathcal{R}(C_k, \mathbb{S} \cup \{\boldsymbol{\delta}\}))| = |\mathcal{R}(C_k, \mathbb{S})| - \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}).$$

We now compute the marginal increment of the data part:

$$\begin{aligned}\Delta L(C_k \mid (\{\boldsymbol{\delta}\} \mid \mathbb{S})) &= L(C_k \mid (\mathbb{S} \cup \{\boldsymbol{\delta}\})) - L(C_k \mid \mathbb{S}) \\ &= |\mathcal{R}(C_k, (S \cup \{\boldsymbol{\delta}\}))| \cdot MC - |\mathcal{R}(C_k, \mathbb{S})| \cdot MC \\ &= \big(|\mathcal{R}(C_k, \mathbb{S})| - \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S})\big) \cdot MC - |\mathcal{R}(C_k, \mathbb{S})| \cdot MC \\ &= -\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) \cdot MC. \end{aligned} \tag{6}$$

Total MDL increment. The total marginal increment when adding $\boldsymbol{\delta}$ to $\mathbb{S}$ is defined as follows:

$$\begin{aligned}&\Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{S})) = L(C_k, (\mathbb{S} \cup \{\boldsymbol{\delta}\})) - L(C_k, \mathbb{S}) = \\ &\qquad \Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{S}) + \Delta L(C_k \mid (\{\boldsymbol{\delta}\} \mid \mathbb{S})) \xRightarrow{(5),(6)} \\ &\Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{S})) = \text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1) \\ &\qquad + (\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) + 1)p_sz - \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) \cdot MC \Rightarrow \\ &\Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{S})) = \text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1) + p_sz \\ &\qquad + \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S})(p_sz - MC) \end{aligned} \tag{7}$$

Supermodularity of $L(C_k, \mathbb{S})$. Let $\mathbb{A} \subseteq \mathbb{B} \subseteq \Delta_k$ and $\boldsymbol{\delta} \in \Delta_k \setminus \mathbb{B}$. Then,

$$\begin{aligned}&\Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{A})) - \Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{B})) = \\ &\quad = \big(\Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{A}) - \Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{B})\big) + \\ &\quad + \big(\Delta L(C_k \mid (\{\boldsymbol{\delta}\} \mid \mathbb{A})) - \Delta L(C_k \mid (\{\boldsymbol{\delta}\} \mid \mathbb{B}))\big), \end{aligned} \tag{8}$$

where

$$\Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{A}) - \Delta L(\{\boldsymbol{\delta}\} \mid \mathbb{B}) \overset{(5)}{=} \big(\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{A}) - \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{B})\big)\, p_sz \tag{9}$$

and

$$\begin{aligned}&\Delta L(C_k \mid (\{\boldsymbol{\delta}\} \mid \mathbb{A})) - \Delta L(C_k \mid (\{\boldsymbol{\delta}\} \mid \mathbb{B})) \\ &\qquad \overset{(6)}{=} -\big(\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{A}) - \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{B})\big) \cdot MC \end{aligned} \tag{10}$$

$$\begin{aligned}(8) \xRightarrow{(9),(10)} &\Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{A})) - \Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{B})) \\ &= \big(\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{A}) - \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{B})\big)(p_sz - MC) \end{aligned} \tag{11}$$

It stands that $\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{A}) - \gamma(\{\boldsymbol{\delta}\} \mid \mathbb{B}) \geq 0$, due to (2), and by the definition of C we have $MC \geq 2p_sz$, hence $p_sz - MC \leq 0$. Therefore (11) implies

$$\begin{aligned}&\Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{A})) - \Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{B})) \leq 0 \Rightarrow \\ &\qquad \Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{A})) \leq \Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{B})). \end{aligned}$$

In terms of the set function $L(C_k, \mathbb{S})$, this is exactly

$$L(C_k, (\mathbb{A} \cup \{\boldsymbol{\delta}\})) - L(C_k, \mathbb{A}) \leq L(C_k, (\mathbb{B} \cup \{\boldsymbol{\delta}\})) - L(C_k, \mathbb{B}),$$

which is the diminishing-increments condition characterizing supermodularity. Hence $L(C_k, \mathbb{S})$ is supermodular on Δ_k. □

Proposition 2 (Submodularity of the MDL reduction). *Given Proposition 1, the MDL reduction $\mathcal{J}(\mathbb{S}) = L(C_k, \varnothing) - L(C_k, \mathbb{S})$ is a monotone submodular set function.*

Proof. **Submodularity of the MDL reduction.** Define the MDL reduction

$$\mathcal{J}(\mathbb{S}) = L(C_k, \varnothing) - L(C_k, \mathbb{S}).$$

Since $L(C_k, \cdot)$ is supermodular, $\mathcal{J}(\cdot)$ is submodular (a constant minus a supermodular function). To see this directly, for $\mathbb{A} \subseteq \mathbb{B}$ and $\boldsymbol{\delta} \notin \mathbb{B}$:

$$\begin{aligned}\mathcal{J}(\mathbb{A} \cup \{\boldsymbol{\delta}\}) - \mathcal{J}(\mathbb{A}) &= \big(L(C_k, \mathbb{A}) - L(C_k, (\mathbb{A} \cup \{\boldsymbol{\delta}\}))\big) \\ &\geq \big(L(C_k, \mathbb{B}) - L(C_k, (\mathbb{B} \cup \{\boldsymbol{\delta}\}))\big) \\ &= \mathcal{J}(\mathbb{B} \cup \{\boldsymbol{\delta}\}) - \mathcal{J}(\mathbb{B}), \end{aligned}$$

where the inequality is exactly the supermodularity inequality proved above. Thus $\mathcal{J}$ is submodular.

Monotonicity of the MDL reduction. Under the admissible addition assumption (Assumption 1), $\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) \geq 1$. By definition of $\boldsymbol{\delta}^*$, for any $\boldsymbol{\delta} \in \Delta_k$ we have

$$\begin{aligned}&\text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1) + 2p_sz \\ &\qquad \leq \text{bits}(\|\boldsymbol{\delta}^*\|_0) + \log_2(\|\boldsymbol{\delta}^*\|_1) + 2p_sz = MC, \end{aligned}$$

and therefore

$$\text{bits}(\|\boldsymbol{\delta}\|_0) + \log_2(\|\boldsymbol{\delta}\|_1) + p_sz \leq MC - p_sz. \tag{12}$$

We also know that $\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) \geq 1$ and $p_sz - MC \leq 0$, hence

$$\gamma(\{\boldsymbol{\delta}\} \mid \mathbb{S}) \cdot (p_sz - MC) \leq p_sz - MC \tag{13}$$

$$(7) \xRightarrow{(12),(13)} \Delta L(C_k, (\{\boldsymbol{\delta}\} \mid \mathbb{S})) \leq (MC - p_sz) + (p_sz - MC) \leq 0.$$

Hence,

$$\mathcal{J}(\mathbb{S} \cup \{\boldsymbol{\delta}\}) - \mathcal{J}(\mathbb{S}) = -\Delta L(C_k, \{\boldsymbol{\delta}\}) \geq 0,$$

so $\mathcal{J}$ is monotone over admissible additions. □

C Algorithms

Algorithm 3 Galactic-G: Optimal

Input: Cluster C_k, groups $\mathcal{G}_k$, budget μ_k
Output: Global perturbation set $\mathbb{S}_{C_k}$

```
procedure Optimal(Δ, C, μ)
    return arg min_{S⊆Δ, |S|≤μ} L(C, S)   ▷ Exhaustive search for small |Δ|
end procedure
```

Algorithm 4 Galactic-G: Greedy

Input: Cluster C_k, groups $\mathcal{G}_k$, budget μ_k
Output: Global perturbation set $\mathbb{S}_{C_k}$

```
procedure Greedy(Δ, C, μ)
    S ← ∅
    while |S| < μ do
        δ* ← arg min_{δ∈Δ\S} (L(S ∪ {δ}) + L(C | S ∪ {δ}))
        if L(S ∪ {δ*}) + L(C | S ∪ {δ*}) ≥ L(S) + L(C | S) then
            break        ▷ Stop if the joint description length ceases to decrease
        end if
        S ← S ∪ {δ*}
    end while
    return S
end procedure
```

D Technical Implementation Details

This section provides a technical report of the model specifications and optimization trajectories required for reproducible local search, alongside the structural mechanics of temporal segmentation for subgroup discovery. Furthermore, we formalize the target selection policies and the mathematical formulation of multi-level importance weighting used to enforce temporal consistency and discriminative precision during counterfactual generation.

D.1 Model Specifications

As surrogate, we use a deep 1D convolutional residual network featuring three blocks with kernel sizes $\{8, 5, 3\}$. Each block employs batch normalization, ReLU activation, and shortcut connections to preserve gradient flow. Global Average Pooling (GAP) aggregates features before the final dense projection.

D.2 Local Search and Reproducibility

The Adam optimizer utilizes a step size of 0.01 and a stagnation threshold at 0.005 to filter negligible perturbations. All experiments were conducted on an AMD Ryzen 9 5950X CPU and an NVIDIA RTX A6000 GPU. For the local baseline comparison, we explain a random sample of 30% of correctly classified instances per cluster to ensure a statistically representative evaluation of local fidelity.

Implementation Details. Temporal sequences are partitioned using NNSegment with a window size of $\lfloor T/10 \rfloor$ and a step size of $\lfloor T/6.67 \rfloor$. We employ K-medoids clustering for group formation, with the optimal cluster count determined via the Silhouette coefficient. Local search utilizes the Adam optimizer ($max_iter = 500$, $patience = 2$). For our MDL objective, we assume a standard 64-bit pointer size and compute log-likelihoods as $\log_2(\cdot + 1)$ to preserve monotonicity. Based on the empirical analysis of local counterfactual robustness (Section 5.1, Section F.1), we adopt the *second possible* cluster—i.e., the most probable alternative under the surrogate—as the default target direction for Galactic-L and Galactic-G. We further employ the *combined* importance weighting strategy, integrating both source- and target-cluster discriminative regions. This configuration consistently yielded higher effectiveness and more stable sparsity–cost trade-offs across comparisons (Section 5.2) and is therefore used throughout all subsequent local and global experiments.

D.3 Target Selection Policies and Evaluation Statistics

We consider three policies for selecting (or not selecting) a target cluster during counterfactual optimization: (i) *second possible*, which selects the cluster with the second-highest posterior probability under the surrogate model; (ii) *random*, which randomly selects a target cluster different from the source and explicitly optimizes toward that cluster; and (iii) *all_random*, which does not predefine a target cluster and instead drives the optimization by pushing the instance away from its current cluster, allowing the search to implicitly discover a new assignment.

Given two policies x and y, and a metric m evaluated over a set of datasets $\mathcal{D}$, we define the mean signed difference:

$$\Delta_m = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \left(m_d^{(y)} - m_d^{(x)} \right).$$

We refer to this quantity as a *Gain* when it improves the metric in the desirable direction (higher is better for success rate; lower is better for cost, sparsity, and runtime), and as a *Loss* otherwise.

D.4 Temporal Segmentation and Subgroup Discovery

Galactic builds on the premise that cluster identity in time series is defined by distinctive temporal topologies. However, unsupervised clusterings often exhibit internal variability, including instances not strictly conforming to a single centroidal pattern (i.e., criticisms [17]). To preserve interpretability while accounting for heterogeneity, we adopt a multi-modal strategy capturing dominant sub-patterns within each cluster C_k.

Instance-wise Partitioning. For each series $x \in C_k$, we compute a segmentation $\mathcal{S}(x) = \{I_1, I_2, \ldots, I_M\}$, where each segment $I_m = [\tau_{m-1} + 1, \ldots, \tau_m]$ is a contiguous interval defined by breakpoints $0 = \tau_0 < \tau_1 < \cdots < \tau_M = T$. Segmentations are obtained via NNSegment [34], which identifies meaningful change-points by detecting local deviations in the signal.

Subgroup Resolution. To uncover structurally distinct temporal regimes within a cluster, we partition the set of gap vectors $\{\gamma(x) : x \in C_k\}$ using K_{seg}-medoids clustering. The number of subgroups is selected by maximizing the Silhouette score. To ensure that the resulting structure is meaningful, we retain only solutions in which all subgroups contain at least two instances; if no such solution exists, we fall back to the highest-scoring partition even if singleton groups are present. The resulting medoids $\mathcal{P}_k = \{\mathcal{S}_1, \ldots, \mathcal{S}_R\}$ define the *dominant segmentation regimes* of cluster C_k, compactly summarizing its internal temporal variability.

D.5 Multi-level Importance Weighting

Given the dominant segmentation regimes, we quantify the contribution of specific temporal segments to cluster identity, ensuring that counterfactual perturbations are directed toward structurally influential regions. For an instance x belonging to a subgroup with segmentation $\mathcal{S}_r$, the importance of each segment $I_m \in \mathcal{S}_r$ is computed via block-wise permutation:

$$\text{imp}(I_m) = \left| a_c - \frac{1}{B} \sum_{b=1}^{B} a_{c,m}^{(b)} \right|, \tag{14}$$

where a_c denotes the surrogate accuracy for the corresponding cluster and $a_{c,m}^{(b)}$ is the accuracy obtained after permuting the values of segment I_m in the b-th perturbation. Segment-level scores are then propagated to a timestep-level mask $w(x) \in [0, 1]^T$.

Figure 2 illustrates the resulting importance masks under different weighting strategies. Depending on the optimization objective, we consider three complementary schemes:

1. Source-Cluster Importance ($w^{(s)}$). This strategy prioritizes preserving the defining temporal structure of the source cluster C_s.

Segment importances are binarized using a quantile threshold, $w_t^{(s)} = \mathbb{1}\{\text{imp}^{(s)}(I_m) \geq Q_{0.75}\}$, effectively preventing the optimizer from perturbing regions that encode the original cluster identity. An example of this behavior is shown in Fig. 2(b), where the counterfactual search explicitly protects perturbing regions that define the original identity.

2. Target-Cluster Importance ($w^{(t)}$). This strategy guides perturbations toward regions that characterize the target cluster C_t. To accommodate the multi-modal structure of C_t, we align all segmentation breakpoints across all subgroup sets of G_t into a unified collection of intervals $[a, b)$. The importance of each unified interval is computed as a weighted average across subgroups:

$$\text{imp}_{[a,b)} = \frac{1}{|G_t|} \sum_{g \in G_t} \sum_{I \in \mathcal{S}_g} \text{imp}_g(I) \cdot \frac{|[a,b) \cap I|}{|I|}. \tag{15}$$

This strategy, visualized in Fig. 2(c), emphasizes temporal regions that must be modified to achieve membership in the target cluster.

3. Combined Importance ($w^{(s,t)}$). The combined strategy simultaneously enforces the structural constraints of the source cluster and the discriminative requirements of the target cluster. We form the unified group set $G_{s,t} = \{g_s\} \cup G_t$, where g_s is the source-group segmentation of x. After aligning all breakpoints from $G_{s,t}$ into unified intervals $[a, b)$, the joint importance is computed as:

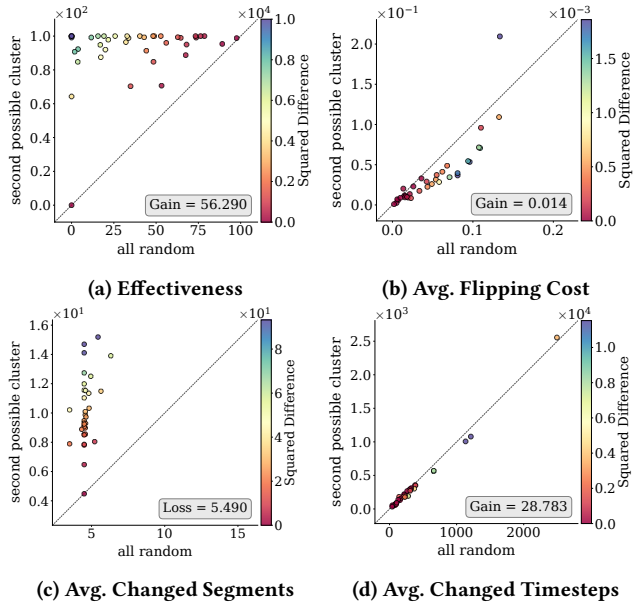
$$\text{imp}_{[a,b)} = \frac{1}{|G_{s,t}|} \sum_{g \in G_{s,t}} \sum_{I \in \mathcal{S}_g} \text{imp}_g(I) \cdot \frac{|[a,b) \cap I|}{|I|}. \tag{16}$$

This strategy is illustrated in Fig. 2(d), where the mask highlights intervals that are simultaneously critical for preserving the source identity and achieving the target assignment.

E Local Baselines and Configurations

In this section, we provide implementation details and configuration choices for all local counterfactual baselines evaluated in Section 5.3.

Baselines. k-Nearest Neighbors (k-NN): This retrieval-based baseline represents the "least effort" strategy for generating a counterfactual explanation. For a series x, we identify the k closest series in the training set belonging to different clusters. A successful counterfactual is found if any of these neighbors are correctly classified into the target cluster C_t by the surrogate model. If successful, no further optimization is performed. TSEvo [19]: A multi-objective evolutionary approach that treats the surrogate model as a black-box. It does not require gradient access and generates counterfactuals through genetic operations. Specifically, it utilizes *mutation* (applying transformations such as noise, interpolation, and shifting) and *crossover* (recombining segments from different series) to navigate the search space. Glacier [40]: The closest baseline to our framework, which follows a gradient-driven search guided by local constraints. Glacier identifies segments and assigns importance scores to discourage changes in certain regions via a binary constraint vector c in its loss function. We evaluate two variants: Glacier-Local (example-specific constraints based on individual explanations) and Glacier-Global (applying a common importance mask derived from the whole dataset).



(a) Effectiveness **(b) Avg. Flipping Cost**

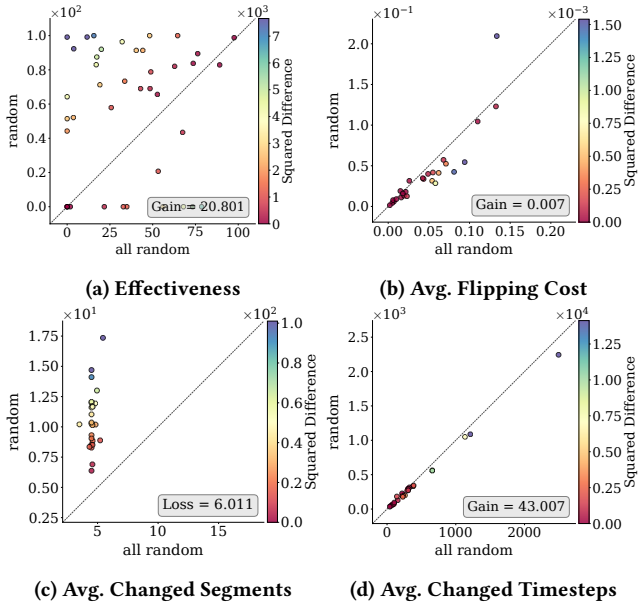
(c) Avg. Changed Segments **(d) Avg. Changed Timesteps**

Figure 5: Galactic-L Target Selection Analysis. Comparison of the second possible and the all random policies across the UCR datasets.

Experimental Configuration and Reproducibility. For a fair comparison, we standardized the search conditions across all methods. For TSEvo, we capped the number of generations at 10 (default: 500), as the default was computationally infeasible for large-scale evaluation on the UCR Archive. Preliminary runs showed that effectiveness stabilized early, with 10 generations already achieving high flipping success while ensuring tractable and reproducible runtimes. For Glacier, we utilized their default hyperparameters but opted for the variant without autoencoders (*NoAE*). This ensures that the generated counterfactuals are evaluated purely on the L_2 flipping cost measure without being biased by the generative preference of the model for "realistic" data distributions, which can intentionally increase edit costs. All methods are evaluated using the same surrogate model and afc (L_2).

F Extended Experimental Results

This section provides a supplementary analytical deep-dive into the empirical performance of the Galactic framework, offering an additional exploration of the hyperparameter space and algorithmic configurations that govern counterfactual generation. We extend the core evaluations presented in the main manuscript by dissecting the nuances of target selection policies and the sensitivity of the global selection mechanism Galactic-G to the perturbation budget μ. Furthermore, we provide granular, per-dataset baseline comparisons for both local and global explanations, ensuring a robust validation of the superiority of our framework across diverse time-series topologies and clustering manifolds.



(a) Effectiveness (b) Avg. Flipping Cost

(c) Avg. Changed Segments (d) Avg. Changed Timesteps

Figure 6: Galactic-L Target Selection Analysis. Comparison of the random and the all random policies across the UCR datasets.

F.1 Search Policy: Target Selection

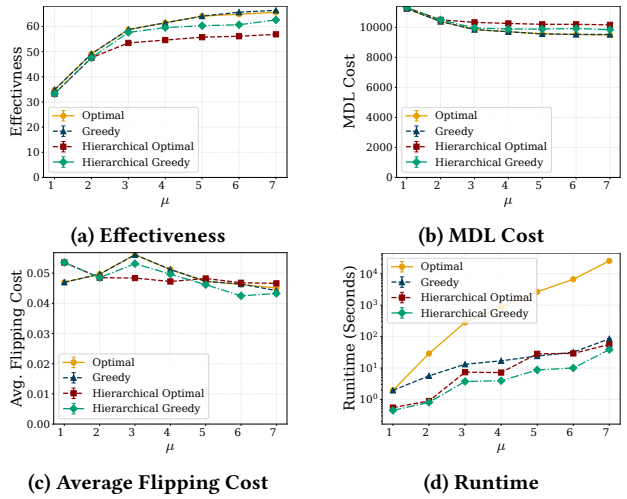
We further analyze target specification by comparing *second possible* and *random* policies against *all_random*, where no explicit target is defined, and optimization is driven by pushing away from the source cluster. Figure 5 compares *second possible* versus *all_random*, while Figure 6 compares *random* versus *all_random* across UCR datasets.

Relative to *all_random*, the *second possible* policy yields a substantial gain in effectiveness, consistently achieving higher success rates. This gain is accompanied by a loss in segment-level sparsity, reflected in a larger number of modified segments, but a gain at the timestep level, with fewer total modified timesteps. This pattern indicates that directing the search toward a specific alternative cluster may require interacting with more semantic regions, while still keeping the actual temporal changes compact. A similar but weaker trend is observed for the *random* policy. Explicitly selecting a target cluster at random improves effectiveness compared to *all_random*, but with smaller gains than *second possible* in effectiveness.

Overall, these results show that explicitly defining a target cluster, particularly when guided by the probability structure of the surrogate, prioritizes successful counterfactual discovery and concentrates perturbations within short temporal intervals, accepting moderate increases in segment-level edits in exchange for substantially higher effectiveness.

F.2 Galactic-G Sensitivity Analysis: Effectiveness vs. Efficiency

Within Galactic-G, we evaluate the trade-off between explanatory quality and efficiency by varying the perturbation budget μ. As shown in Section F.2, which illustrates this analysis for the



(a) Effectiveness (b) MDL Cost

(c) Average Flipping Cost (d) Runtime

Figure 7: Galactic-G selections with varied μ

ECG200 dataset, increasing μ yields a non-linear, monotonic improvement in effectiveness. Initially, the **Minimum Description Length (MDL)** objective prioritizes high-impact, parsimonious perturbations that maximize cluster effectiveness while minimizing aggregate flipping costs, with the minimum description length. For this case, results plateau once μ exceeds the intrinsic structural limit of three groups, suggesting a singular, well-optimized perturbation per group sufficiently characterizes the cluster manifold, rendering further counterfactuals redundant.

Furthermore, our analysis validates that while the **hierarchical optimal** achieves local optimality within discrete subgroups, it remains globally sub-optimal when evaluated at the aggregate cluster level. This global sub-optimality arises because the hierarchical decomposition lacks the holistic search visibility required to evaluate cross-group interactions and unified description costs across the entire cluster partition. Regarding efficiency, the **greedy selection** mechanism offers a computationally superior alternative to the optimal search, delivering comparable effectiveness with significantly reduced runtime. While the **hierarchical greedy** variant is the most efficient for small budgets, the computational overhead associated with propagating search operations across numerous subgroups causes its complexity to converge with the standard greedy approach as μ increases. These findings confirm that the Galactic effectively balances the combinatorial challenge of global selection with actionable, high-fidelity explanations.

F.3 Additional Results for Local Evaluation

Table 5 presents additional results for 20 UCR datasets.

F.4 Additional Results for Global Evaluation

Table 6 presents additional results for 20 UCR datasets.

Table 4: Metadata of the selected UCR time-series datasets and Surrogate classification accuracies.

Dataset	Instances	Length	Clusters	Type	ResNet Acc.	Dataset	Instances	Length	Clusters	Type	ResNet Acc.
ArrowHead	211	251	3	IMAGE	73.81%	ItalyPowerDemand	1096	24	2	SENSOR	95.45%
Lightning7	143	319	7	SENSOR	75.86%	CBF	930	128	3	SIMUL.	100.00%
Car	120	577	4	SENSOR	91.67%	MiddlePhalanxTW	553	80	6	IMAGE	58.56%
Coffee	56	286	2	SPECTRO	100.00%	Computers	500	720	2	DEVICE	92.00%
CricketZ	780	300	12	HAR	84.62%	PowerCons	360	144	2	DEVICE	82.61%
Earthquakes	461	512	2	SENSOR	75.27%	Rock	70	2844	4	SPECTRO	78.57%
ECG5000	5000	140	5	ECG	94.50%	ShapeletSim	200	500	2	SIMUL.	100.00%
ECGFiveDays	884	136	2	ECG	100.00%	ToeSegmentation2	166	343	2	SENSOR	97.06%
FaceFour	112	350	4	IMAGE	100.00%	TwoLeadECG	1162	82	2	ECG	100.00%
FordB	4446	500	2	SENSOR	90.00%	Wine	111	234	2	SPECTRO	47.83%
Ham	214	431	2	SPECTRO	86.05%	Yoga	3300	426	2	IMAGE	96.67%
InlineSkate	650	1882	7	MOTION	52.85%	Wafer	7164	152	2	SENSOR	99.93%
ACSF1	200	1460	10	DEVICE	80.00%	ECG200	200	96	2	ECG	90.00%
Fish	350	463	7	IMAGE	95.71%	SyntheticControl	600	60	6	SIMUL.	100.00%
UMD	180	150	3	SIMUL.	100.00%	Worms	258	900	5	MOTION	62.22%

Table 5: Additional Local Performance Benchmarking. Comparative analysis of Galactic-L against kNN, TSEvo, and Glacier variants across effectiveness, average flipping cost, average segments changed, average timesteps changed, and runtime. The null indicator "-" is assigned throughout the metrics where the corresponding effectiveness is zero.

Dataset	KNN					TSEvo					Glacier-L					Glacier-G					Galactic-L				
	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT
Computers	100	0.1	29.83	720	0.41	100	0.07	29.83	720	906.87	35.33	0	29.68	640.91	31.52	36	0	29.72	648.85	34.47	87.33	0.01	8.53	399.92	29.42
CricketZ	100	0.09	30.22	300	1.01	100	0.1	30.22	300	2366.73	11.97	0.01	32.11	300	22.94	14.53	0.01	32.97	300	26.57	99.57	0.01	8	190.56	75.75
Earthquakes	100	0.24	39.67	512	0.37	100	0.19	39.67	512	1289.46	31.65	0.01	41.41	509.32	27.06	34.53	0	41.56	509.77	29.7	94.24	0.03	10.97	294.5	65.15
ECG5000	100	0.07	20.15	140	7.12	100	0.14	20.15	140	9583.36	3.27	0.01	26.51	133.78	28.94	5.13	0.01	24.45	128.92	44.97	61.8	0.01	7.43	50.2	282.68
ECGFiveDays	100	0.11	22.79	136	1.47	100	0.09	22.79	136	1737.3	22.93	0.01	22.44	132.8	36.41	52.63	0.02	22.51	129.69	78.49	100	0.03	5.94	36.84	121.5
FaceFour	100	0.2	29.5	350	0.19	100	0.21	29.5	350	339.1	17.65	0.02	29.33	350	4.67	5.88	0.01	31	350	1.46	100	0.04	8.91	242.62	16.06
FordB	100	0.14	30.1	500	16.04	99.93	0.08	30.11	500	23812.4	23.31	0	44.1	498.73	287.05	27.66	0	44.07	498.78	353.71	62.44	0.01	11.32	244.14	104.02
GesturePebbleZ2	100	0.09	28.43	451.2	0.41	100	0.16	28.27	450.32	545.86	0	-	-	-	0	-	-	-	-	0	100	0.01	7.41	321.45	141.58
GunPoint	100	0.14	27.53	150	0.27	100	0.1	27.53	150	333.07	45	0.02	28.85	149.96	17.55	48.33	0.02	28.83	150	30.31	96.67	0.06	7.53	62.55	13.07
Ham	100	0.16	33.98	431	0.09	100	0.14	33.98	431	379.56	50.77	0.01	33.85	420.39	22.04	61.54	0.01	34.88	416.38	38.64	98.46	0.03	8.97	280.91	8.61
InlineSkate	100	0.06	23.57	1882	0.94	100	0.1	23.57	1882	1469.5	68.11	0	23.5	1566.22	56.12	65.41	0	23.43	1552.45	103.88	98.92	0.01	6.51	1146.64	34.79
ItalyPowerDemand	0	-	-	-	0.08	100	0.1	6.89	24	2821.62	8.51	0.02	8.32	24	15.33	15.81	0.01	8.38	24	29.17	73.86	0.05	2.17	15.49	71.84
Lightning7	100	0.14	26.58	319	0.09	100	0.15	26.58	319	141.8	0	-	-	-	-	6.98	0.01	27	319	3.2	95.35	0.04	7.32	169.46	27.52
MiddlePhalanxTW	100	0.07	17.93	80	0.25	100	0.14	17.93	80	533.81	60.24	0.01	18.35	79.99	56.28	69.88	0.01	18.22	79.99	90.65	100	0.05	4.86	43.48	26.34
PowerCons	100	0.21	38.68	141.57	0.09	100	0.15	38.71	142.94	213.43	13.04	0.01	43.22	144	6	8.7	0.01	43.67	144	6.35	86.96	0.04	11.33	58.82	18
ShapeletSim	100	0.38	26.52	500	0.08	100	0.28	26.52	500	194.66	0	-	-	-	-	0	-	-	-	0	63.33	0.11	9.82	264.89	29.9
ToeSegmentation2	100	0.2	26.94	343	100	0.17	26.94	343	14	0.01	29.43	343	10	0.01	29	343	100	0.06	7.06	129.7	0.07	160.02	3.29	3.37	51.48
TwoLeadECG	100	0.08	22.61	82	0.47	100	0.08	22.61	82	1084.46	12.89	0.02	26.2	82	28.89	49.57	0.02	24.01	81.99	166.6	91.98	0.04	6.49	18.69	76.35
Wafer	0	-	-	-	0.21	100	0.02	18.16	151.99	16581.7	0	-	-	-	0.11	0	-	-	-	0.09	81.44	0.01	5.43	32.87	257.01
Yoga	0	-	-	-	0.33	100	0.04	24.77	426	9700.39	48.08	0	24.45	423.7	273.55	0	-	-	-	11.02	97.88	0.01	6.64	219.55	223.98

Table 6: Additional Global Performance Benchmarking. Comparative analysis of Galactic-G variants against GLOBE-CE* and Glacier-G* across effectiveness, average flipping cost, average segments changed, average timesteps changed, and runtime. The * indicates adaptations for the global timeseries clustering context; the null indicator "-" is assigned throughout the metrics where the corresponding effectiveness is zero.

Dataset	Galactic-G																				Competitors									
	Optimal					Greedy					Hierarchical					Hierarchical Greedy					GLOBE-CE*					Glacier-G*				
	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT	eff	afc	asc	atc	RT
Computers	36.9	0.006	5.0	580	303.33	36.9	0.006	5.0	580	25.60	34.7	0.007	5.0	577	17.03	34.9	0.006	5.0	579	8.12	42.1	0.012	7.8	720	218.32	8.2	0.003	12.5	720	15.24
CricketZ	29.4	0.014	3.6	178	145.44	29.0	0.012	3.8	175	10.90	15.6	0.009	4.1	167	6.35	15.7	0.009	4.1	168	5.09	85.8	0.032	6.4	275	29.49	4.4	0.006	5.8	250	25.44
Earthquakes	50.1	0.055	3.7	241	303.97	50.1	0.055	3.7	241	30.57	36.1	0.029	4.0	233	20.74	40.3	0.034	3.9	234	15.52	21.3	0.034	7.0	512	139.45	3.1	0.005	7.0	512	20.94
ECG5000	48.3	0.018	4.6	37	778.67	48.1	0.019	4.6	37	41.15	44.7	0.018	4.6	36	47.73	45.2	0.017	4.5	35	31.83	75.7	0.106	8.4	140	175.61	0.2	0.000	3.0	27	0.00
ECGFiveDays	82.6	0.038	3.1	38	5367.02	82.6	0.038	3.1	38	65.79	66.5	0.034	3.1	37	50.92	66.4	0.036	3.1	39	20.98	50.0	0.138	7.2	136	91.62	21.3	0.019	7.6	136	0.00
FaceFour	50.6	0.037	3.8	178	271.77	50.6	0.032	3.5	174	6.68	25.0	0.032	3.1	169	6.56	25.0	0.030	3.0	168	4.10	55.3	0.059	5.2	262	40.68	0.0	-	-	-	20.83
FordB	10.7	0.013	3.4	142	25204.26	10.7	0.013	3.4	142	590.80	1.8	0.007	2.6	149	324.76	1.8	0.006	2.6	149	143.58	30.5	0.021	7.0	500	1596.26	1.0	0.003	7.0	499	18.00
GesturePebbleZ2	53.5	0.011	4.4	296	37.23	53.5	0.011	4.4	296	25.07	42.2	0.008	4.7	283	22.59	42.5	0.007	4.7	285	22.58	15.0	0.002	6.0	303	33.40	0.0	-	-	-	18.12
GunPoint	76.5	0.109	3.3	56	119.67	76.5	0.094	3.3	56	13.30	65.5	0.135	3.7	60	13.20	65.5	0.078	3.6	60	7.53	57.0	0.277	7.0	150	36.44	18.5	0.023	7.0	150	12.39
Ham	51.8	0.034	4.7	259	226.36	51.8	0.034	4.7	259	15.22	50.3	0.033	4.5	245	18.96	50.3	0.033	4.5	245	7.26	5.7	0.069	7.5	431	61.08	8.1	0.010	12.1	431	19.55
InlineSkate	89.1	0.007	4.4	657	481.00	89.1	0.006	4.1	614	15.33	86.8	0.007	4.6	814	5.57	86.8	0.006	3.9	628	4.40	85.8	0.008	7.0	1882	218.86	27.7	0.001	6.0	1603	20.08
ItalyPowerDemand	49.8	0.064	2.3	12	259.87	49.8	0.059	2.3	11	26.30	48.8	0.076	2.2	11	1.75	49.6	0.057	2.4	12	6.87	70.6	0.097	3.2	24	55.17	2.8	0.012	1.7	12	0.00
Lightning7	60.5	0.055	4.0	145	36.24	60.5	0.050	3.9	140	6.85	25.8	0.021	3.3	110	6.00	26.6	0.018	3.1	102	5.86	36.7	0.033	4.0	182	33.41	8.0	0.006	4.0	182	9.35
MiddlePhalanxTW	86.5	0.046	3.5	43	7.40	86.5	0.045	3.3	42	3.55	68.3	0.042	3.4	41	1.73	68.3	0.042	3.4	41	1.72	100.0	0.133	7.0	80	27.94	41.1	0.014	7.0	80	10.71
PowerCons	37.7	0.048	6.1	57	1548.78	37.7	0.048	6.1	57	29.43	30.5	0.040	6.0	53	49.89	37.7	0.048	6.1	57	26.99	47.2	0.141	10.3	144	46.46	2.3	0.011	24.7	144	15.96
ShapeletSim	45.5	0.135	5.5	247	110.65	45.5	0.112	5.5	245	14.05	45.5	0.135	5.5	247	11.68	45.5	0.112	5.5	247	9.70	0.0	-	-	-	74.31	0.0	-	-	-	18.25
ToeSegmentation2	67.2	0.084	4.9	142	173.25	67.2	0.084	4.9	142	22.74	61.6	0.060	5.1	143	22.71	61.6	0.060	5.1	143	17.82	22.7	0.033	7.8	343	46.68	3.6	0.013	16.9	343	16.57
TwoLeadECG	39.7	0.035	3.3	28	1683.43	39.7	0.034	3.3	28	60.38	13.8	0.032	3.3	28	66.77	33.7	0.034	3.4	28	34.42	62.7	0.127	7.0	82	71.97	20.6	0.015	7.0	82	9.27
Wafer	38.3	0.018	1.3	10	1158.02	38.3	0.018	1.3	10	166.75	28.1	0.011	1.3	10	131.90	28.1	0.011	1.3	10	72.22	50.0	0.006	3.5	76	672.45	3.9	0.002	7.0	152	0.00
Yoga	48.6	0.007	5.9	230	38593.63	48.6	0.007	5.9	230	416.69	45.5	0.007	5.9	217	1001.63	47.5	0.007	5.9	218	225.89	57.7	0.036	7.3	426	1043.10	5.9	0.002	8.5	426	17.18